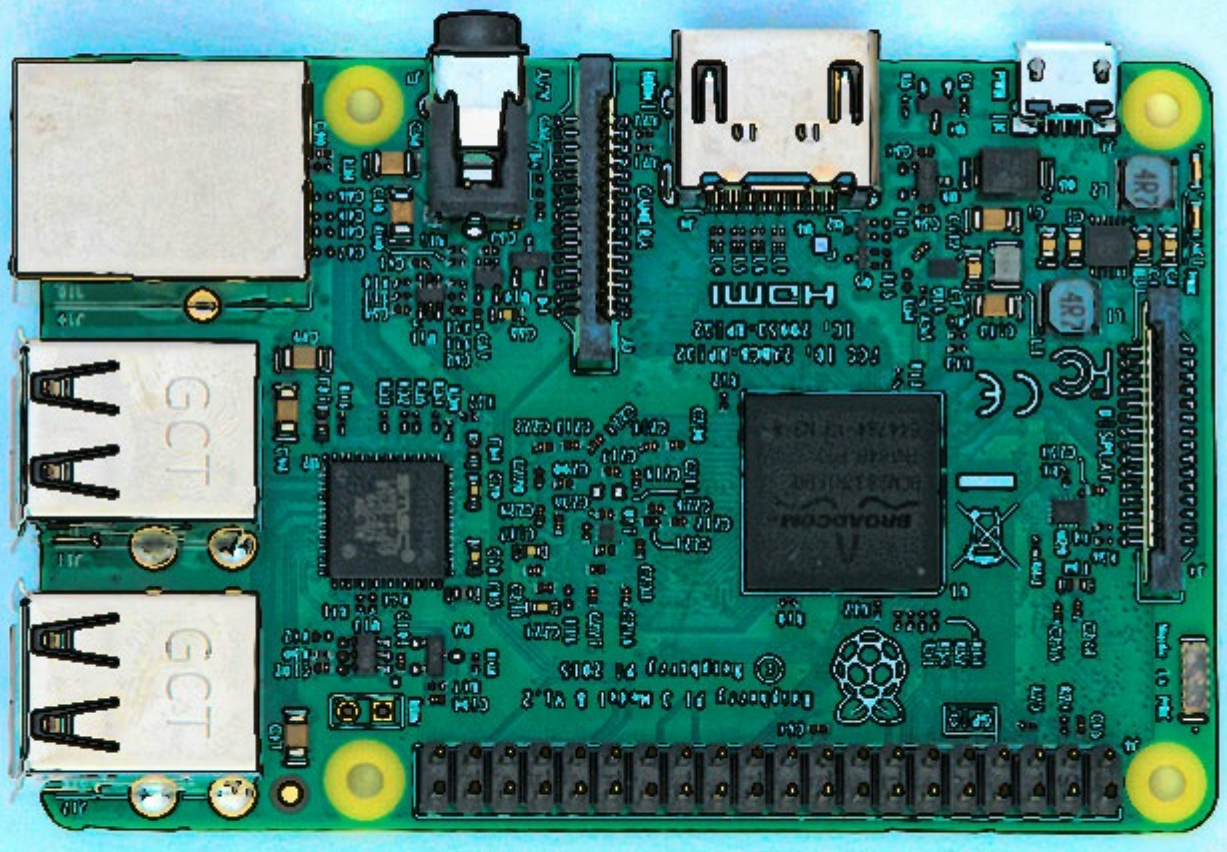
Einar Krogh

An Introduction to the Internet of Things



bookboon
The eBook company

EINAR KROGH

AN INTRODUCTION TO THE INTERNET OF THINGS

An Introduction to the Internet of Things
1st edition

ISBN 978-87-403-3224-7
Peer review by: Professor Øystein Haugen and doctoral research fellow Marius Geitle, Høgskolen i Østfold

CONTENTS

INTRODUCTION

This book is written for introductory courses on embedded systems and the Internet of Things (IoT).

The book consists of two parts. The first part introduces embedded systems. Central topics are sensors, actuators, and the architecture of embedded systems.

The second part of the book is about the IoT. It introduces the IoT, the architecture of the IoT, and important IoT technologies. Other topics are also included, such as cloud computing and big data in connection with the IoT.

In the time to come, the IoT will have an increasing impact on our lives. The IoT is also very popular. Many predict that the IoT will be the next big digital revolution.

I would like to thank Professor Øystein Haugen and Doctoral Research Fellow Marius Geitle for reading the manuscript of the book and for many suggestions.

PART 1 EMBEDDED SYSTEMS

1 INTRODUCING EMBEDDED SYSTEMS

1.1 WHAT IS AN EMBEDDED SYSTEM?

An embedded system is a combination of hardware and software to perform a specific task. Allowing the software to control the hardware provides an opportunity for intelligent behavior and smart solutions.

An embedded system consists of three main components:

- Hardware
- Application software
- Operating system

An operating system is computer software that manages hardware and other software. While hardware and software are always included in an embedded system, there are exceptions when it comes to the operating system. Very simple embedded systems do not always have an operating system.

Most embedded systems do not have a user interface for humans, but some may have a kind of user interface, such as a touch screen. There are also embedded systems that have quite complex user interfaces, such as mobile phones.

Software for embedded systems is usually referred to as firmware. Instead of storing data on a hard drive as in computers, individual programs in an embedded system are normally stored on a chip and called firmware.

Figure 1.1 The development of embedded systems has become much easier since the introduction of two platforms: Arduino and Raspberry Pi.
The figure shows a Raspberry Pi 3 Model B.

Electronic equipment designed for the engineering market is classified as embedded systems. An embedded system is an electronic system that can be programmed to operate and organize one or more tasks. Embedded systems are an important part of today's electronic industry.

1.2 REAL-TIME EMBEDDED SYSTEMS

Real-time systems are computer systems that monitor, respond to, or control an external environment. The external environment is connected to an embedded system that often has sensors, actuators, and other interfaces.

A real-time system must be able to respond to events in an external environment as soon as they happen. Another name for real-time systems is reactive systems because their primary purpose is to respond to events in the environment.

An example in which a real-time system is necessary is a system with the task of reducing the speed of a car when there is something on the road. Then the car must brake immediately (in real time) if an accident is to be avoided.

There are two important features of real-time embedded systems:

- Real-time embedded systems must perform flawlessly accurate calculations for events.
- Response to events must occur very quickly during a predefined time interval.

In real-time systems in which real-time calculations are required with accurate results to be delivered within a short time span, the operating system usually plays an important role. With the growing complexity of the hardware in embedded systems, there is a need for an operating system that meets the system requirements and does not miss deadlines.

Household systems for monitoring and control of devices
Systems for cars, subways, aircraft, railways, and ships
Traffic control for motorways, airspace, railway tracks, and ship routes
Medical systems for radiation therapy and patient monitoring
Military applications such as firearms, tracking, command, and control
Robot production systems
Telephone, radio, and satellite communication
Computer games

Figure 1.2 Some examples of real-time systems.

Real-time embedded systems are everywhere. Today's systems range from regular car management systems and kitchen appliances to control systems for air traffic, military weapons facilities, and production line control, including robotics and automation.

1.3 EMBEDDED SYSTEMS COMPARED TO COMPUTERS

Unlike computers that can be used to do many different tasks, embedded systems are most often designed to perform only one particular task, for example, regulating a traffic light. Many embedded systems are also real-time systems, which ordinary computers are not.

Since embedded systems have only one specific purpose, they are more limited in hardware and software functionality than computers. In the case of hardware, this may mean less processing power, less power consumption, less memory, less hardware functionality, and so on. In terms of software, this limitation may mean fewer or simpler applications and no or a small operating system.

Since embedded systems focus only on a particular task, they can be made cheaper and more efficient than computers.

Desktop computer	Embedded system
Runs different programs at different times depending on the needs of the user.	Runs a single, dedicated application at all times.
Has large amounts of (RAM) memory and disk space; both can be readily and cheaply expanded if required.	Has sufficient memory but no excess. Adding more memory is difficult.
All PCs have an essentially identical hardware architecture and run identical software. Software is written for speed.	Embedded systems are highly variable with different CPUs, peripherals, operating systems, and design properties.
Boot-up time may be measured in minutes, and the operating system is loaded from disk and initialized.	Boot-up is almost instantaneous, measured in seconds.

Figure 1.3 The table shows some differences between embedded systems and computers.

Focusing on the differences between computers and embedded systems can give a description of what an embedded system is. A simple description of an embedded system is as follows:

- An embedded system is any data system found in hardware equipment that is not a computer.

1.4 FEATURES OF EMBEDDED SYSTEMS

An embedded system is a system that has a specific purpose and performs either a single or only a few operations. An embedded system is often hardware equipment without any user interface.

Typical properties of embedded systems are low power consumption and low cost. This often results in limited processing possibilities.

An embedded system is a system that demands high quality and reliability. Embedded systems are often used in hardware that must function flawlessly for many years.

Some embedded systems demand very high quality and reliability. For example, if the technical equipment used for an operation in a hospital breaks down, this could have life-threatening consequences, or if a system that controls a car fails on the highway, this can lead to an accident.

Requirements	
Processing power	Microcontrollers or microprocessors control embedded systems.
Operating system	The embedded operating system must be reliable and capable of running with restrictions on memory, size, time, and processing power.
Memory	Computer programs designed for embedded systems are handled as firmware and are stored in ROM or on flash memory chips.
Power consumption	Power consumption is an important factor for any embedded battery-powered system. It is the amount of power consumption that determines the life of the battery.
Flexibility	Flexibility is the ability to change system functionality without extra costs. Software is considered flexible if it can be updated with a new version at any time.
Size	An embedded system should preferably be as small as possible.
Reliability	An embedded system should be highly reliable for achieving good performance over its lifetime.
Security	If a system failure occurs, there should be no damage to the components.
Maintenance	It is important that an embedded system can be repaired or replaced as soon as possible, that is, within a specified time interval.

Figure 1.4 Common features of embedded systems.

1.5 ROBOTS AND EMBEDDED SYSTEMS

A robot is a mechanical and programmed device that performs tasks to help people.

In the old days, robots were controlled by large and expensive computer systems. Many of these robots were stationary because they were too heavy to be carried around. Mobile robots had to be connected to a computer via cables or wireless, but connection and response speed was problematic. The development of embedded systems has had great significance for robots because it has solved this problem.

Figure 1.5 Example of an embedded system. The GoPiGo is a mobile robot car controlled by a Raspberry Pi card. The Raspberry Pi card contains program code controlling the robot's movements (Dexter Industries).

A robot consists of three main components:

1. Sensors that provide feedback from an external environment
2. A mechanical unit (actuator) that can perform actions on the environment
3. An embedded system for communicating between a mechanical device and the sensory data

Very few robots look like the metallic robots in science fiction. Any mechanism that has actuators, sensors, and a controller can be classified as a robot. The definition also includes units that we would not regard as robots, such as the block-free braking system of modern cars.

Some types of robots	Use
Industrial robots	Handling, welding, and inspection of materials, as well as improving productivity. Laboratory applications.
Collaborative robots	A collaborative robot is a robot that can safely and effectively interact with human workers while performing simple industrial tasks.
Mobile robots	Robots that move around on legs, tracks, or wheels.
Educational robots	Educational robots feature learning platforms that are geared to teaching robotics to children, students, and amateurs.
Domestic robots	Vacuum cleaners, floor-washing robots, ironing robots, etc.
Military robots	Use of robots for military combat. Smart missiles and autonomous bombs can be considered robots.

Figure 1.6 The table shows some types of robots.

2 USE OF EMBEDDED SYSTEMS

Embedded systems are found everywhere. For example, they can be found in washing machines, microwave ovens, and household appliances. Other examples are digital watches, toys, traffic lights, industrial robots, and agricultural machinery.

Embedded systems are used in consumer electronics, including mobile phones, video game consoles, digital cameras, GPS receivers, and printers. Home automation uses embedded systems to control light, sound, climate, security, and monitoring.

Embedded systems are used in transportation, fire protection, medical applications, and life-critical systems, as these systems can be isolated from hacking and are therefore more reliable. For fire protection, the embedded systems can be designed to have a greater ability to withstand higher temperatures and will therefore continue to function if there is a fire.

Aircraft and car transport systems are increasingly using embedded systems. Modern aircraft include advanced technology that also has high-security requirements.

2.1 EMBEDDED SYSTEMS IN VEHICLES

The use of advanced embedded systems in automobiles has increased rapidly in the past two decades. It is expected that by 2020, 90% of automobiles will be connected to the Internet.

Every year, automobile manufacturers pack more embedded systems into their cars for different functionalities, such as ignition, security, and audio systems. The aim is to make the vehicle more efficient and safer.

We shall look at some applications of embedded systems in cars.

Adaptive Cruise Control

If someone in the late 1900s had told us that there would soon come a new technology that would end car accidents, no one would have believed it. However, the situation is that embedded systems that support the driver make driving much safer.

Many modern cars have an embedded adaptive cruise control (ACC) system. ACC is a control system for road vehicles that automatically adjusts the vehicle speed to maintain a safe distance from vehicles ahead. It determines the car's speed using a braking system that takes into account the distance between the vehicle it is in and the vehicles in front. Such cars are usually equipped with radar or LIDAR (light detection and ranging) to determine distance.

Lane Centering

Lane centering, also known as auto steer, is a mechanism designed to keep a car centered in the lane, relieving the driver of the task of steering. Together with ACC, this feature may enable driving without a driver.

Airbag Control System

All modern cars have airbags to make driving safer. In order to inflate the airbags at the right time, you have an airbag control system. It detects a collision using a crash sensor and sends a signal to the airbags so that they are inflated. The entire process from start to finish takes 0.1 seconds.

Anti-Lock Braking System (ABS)

An anti-lock braking system (ABS) is designed to control vehicle braking in a way that gives less chance of slipping on slippery roads. An ABS operates by preventing the wheels from locking up during braking, thereby maintaining tractive contact with the road surface. It ensures better contact with the road by controlling brake pressure if a car begins to slip while braking.

Other Uses of Embedded Systems in a Car

Other uses of in-vehicle systems include the embedded navigation system, electronic stability control (ESCESP), the traction control system (TCS), tire air pressure control, automatic four-wheel drive, and electronic fuel injection.

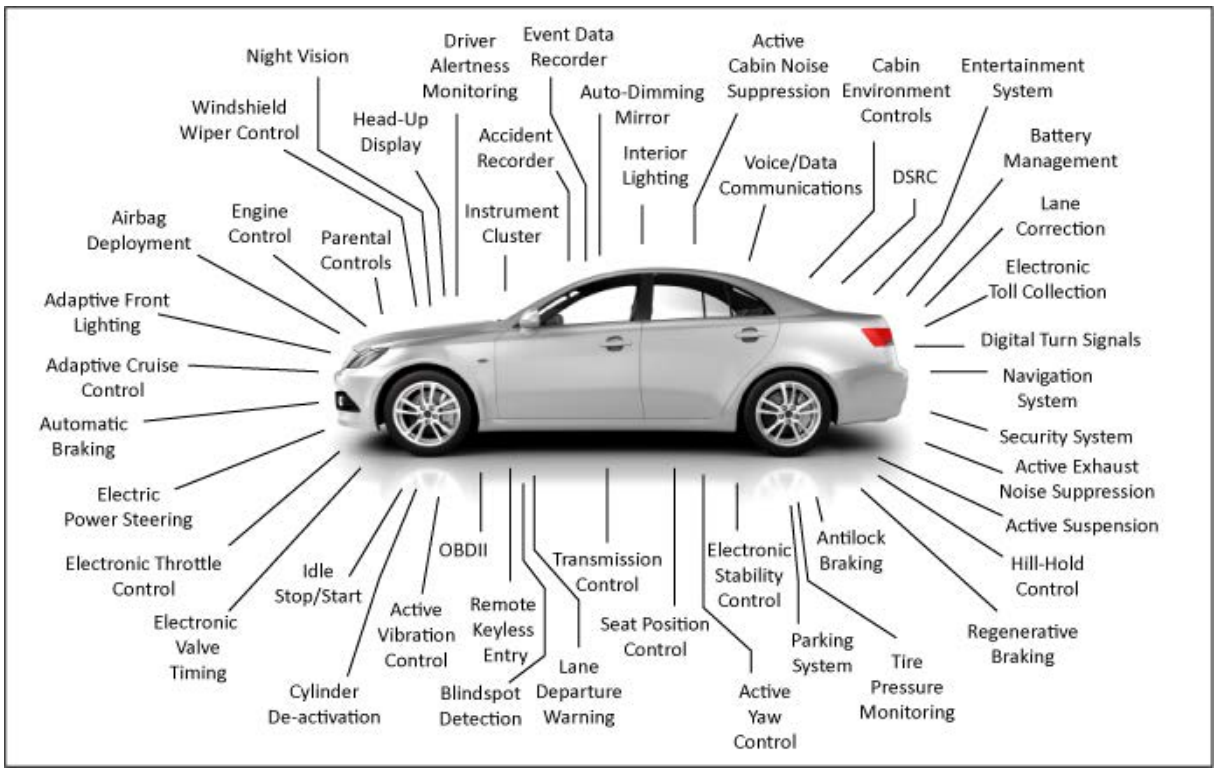


Figure 2.1 Today's automobiles contain dozens of computer chips (Vintage Computer).

2.2 SOME EXAMPLES OF EMBEDDED SYSTEMS

Embedded systems have become part of our daily lives. Figure 2.2 lists some examples of the use of embedded systems.

Washing machines and dishwashers
Lighting systems
Refrigerators and freezers
Vacuum cleaners
Mobile phones
Smartwatches and digital watches
Air conditioners and thermostats
Electric cookers and coffee machines
Electronic parking meters and parking services
CD, iPod and MP3 players
Home security systems
Fire alarms and carbon monoxide detectors
Printers, copiers, fax machines, and scanners
Digital cameras
Electronic safes
GPS navigation devices
Heart rate monitors and pacemakers
Wi-Fi routers
Electronic toys

Figure 2.2 Examples of the use of embedded systems.

3 SENSORS AND ACTUATORS

A sensor is a device that detects and responds to some type of input from the physical environment. Sensors sense; that is, they act as the eyes and ears of an embedded device and detect changes in the environment around them.

Actuators are devices that accept a control command and produce a change in the physical system by generating force, motion, heat, flow, and so forth. You may say that actuators are the hands of embedded systems.

Both sensors and actuators have many practical applications.

3.1 TRANSDUCERS

A transducer is any physical device that converts one form of energy into another. Transducers that convert physical quantities into mechanical quantities are called mechanical transducers. Transducers that convert physical quantities into electrical quantities are called electrical transducers.

The following are some examples of the actions of transducers:

- An electric motor converts electricity into mechanical energy or motion.
- A speaker converts electrical signals into sound waves.
- An incandescent lamp produces light by converting electrical energy into optical energy.
- A solar cell converts light into electricity.

Transducers enable technical equipment to interact with the physical environment. A transducer must therefore have a processing unit and a communication interface.

A sensor is a kind of transducer. A sensor transforms a physical phenomenon into an electrical impulse that can be used in a technological system. For example, a microphone is a transducer (sensor) converting sound waves into electrical signals.

3.2 SENSORS

Sensors are key components of the IoT. They are important in the work of monitoring processes, measurements, and data collection.

A sensor is a device that detects events or changes in the environment and that sends information about them to another location, for example, a server or a web page. Most sensors are designed to measure a physical quantity and then to transform it into a digital value that can be read by humans or used by some data systems.

The sensors are the reason why the IoT is constantly growing. There are many different types of sensors on the market, and they are used for various purposes that encompass all aspects of human life.

Sensor technology is being developed faster and faster because of new discoveries in materials and nanotechnology. The result is increased accuracy, reduced size and cost, and the ability to measure or find values that have not previously been possible to capture. In fact, the sensor technology is developing so rapidly and becoming so advanced that we will see billions of new sensors in production annually within a few years.

The low price of most sensors helps keep the IoT expenses down and enables the use of embedded systems on a large scale. The IoT has changed the manufacturing industry, and sensors are central to the use of the IoT.

Use of Sensors

Sensors are widely used, and there are hundreds of different types of sensors. We have temperature sensors, flow sensors, voltage sensors, humidity sensors, and so on.

For example, autonomous vehicles are full of sensor technology and have sensors to measure power, load, torque, motion, speed, displacement, position, vibration, and shock.

Sensors used in production equipment in a factory can help identify bottlenecks in a manufacturing process. By tackling bottlenecks, factories can reduce waste of production time.

Instead of standard preventive maintenance, which means machine maintenance, predictable maintenance involves using sensors to predict quite accurately when machines need maintenance.

Sensors and actuators must be reliable. In complex embedded systems, a fault in a sensor or an actuator can trigger catastrophic events. Detection of sensor and actuator errors can be difficult, and it affects the performance of critical systems.

Storage of Data From Sensors

Deciding whether to store data from sensors locally or in the cloud is often a dilemma. There are advantages and disadvantages of both options.

Speed is one of the main advantages of local storage. Storing data on external hard drives is faster than uploading to the cloud. You also have full control of your backups, which means better control of who accesses your data. Disconnecting the drives from the network protects your data against attacks.

There are many advantages of backing up sensor data to the cloud. For one, it is cost-effective, and maintenance is not an issue as cloud storage providers handle all the upgrades and troubleshoot any issues that might arise.

Another advantage of cloud storage is scalability. When you need to increase storage space, it is as simple as notifying your service provider. You can increase or decrease space as needed. Should a disaster occur on-site, your data will remain safe. Securing data remotely means that you do not have to worry about losing backups of your data.

Accessibility is also a plus of cloud storage. Data stored in the cloud is easily accessed on any device that has an Internet connection. You can log into your cloud account, and your data is there when you need it.

A disadvantage of cloud storage is security and privacy. There are concerns with valuable and important data being stored remotely. Before adopting cloud technology, you should be aware that you are giving sensitive information to a third-party cloud service provider, and this could potentially be a risk.

Another disadvantage is lifetime costs. With public cloud storage, the price costs might increase over the years and tend to add up. At this point, the lifetime costs will hit you. If your applications are local and your data is in the cloud, then it can add to networking costs.

3.3 SOME SENSORS

Sensors perform very different tasks, and each IoT system requires a specific type of sensor. We will look at some sensors and their use.

Temperature Sensors

This widely used sensor type measures the temperature or the heat of a given medium. There are several types of temperature sensors on the market.

Temperature sensors are used in everything from simple thermostats to highly sensitive semiconductors that are capable of controlling complex processes.

Proximity Sensors

A proximity sensor is a sensor capable of determining the distance to a nearby object without having physical contact with the object. Proximity sensors use electromagnetic radiation or radar to detect movements or obstructions.

Proximity sensors are good at detecting movement. They are a common component of equipment that involves safety, security, or efficiency. These sensors are therefore used in vehicles to detect obstacles in front of the vehicle when in motion.

Proximity sensors are used in stores. Retailers use proximity sensors in their stores to investigate which goods customers get most close to and are most interested in. The data is processed and sent to the retailers' mobile phones.

Proximity sensors can be used in parking systems, museums, airports, etc. For example, parking sensors are proximity sensors designed for vehicles to alert the driver of obstacles when parking. A ground proximity warning system (GPWS) is a system to alert pilots if their aircraft is in immediate danger of flying into the ground or an obstacle.

Motion Detector Sensors

A motion detector is an electronic device used to detect physical motion in an area and to transform this motion, both movement of an object and movement of people, into an electrical signal.

Motion detection plays an important role in the security industry. Companies use these sensors in areas where there never should be any movement at all. With these sensors installed, it is then easy to notice if anyone or anything is moving. They are used, for example, for intrusion detection, door control, automatic parking systems, automated sinks, toilet fans, hand dryers, and automated lighting.

Although their primary use is now in the security industry, the number of possible applications of these sensors will only grow as the use of motion detection technology develops.

LIDAR

Light detecting and ranging (LIDAR) is a type of sensor that measures the distance to a target by measuring a laser pulse reflection on the target.

LIDAR broadcasts laser energy. As a laser hits an object, some of the energy will be reflected back toward the LIDAR transmitter. This type of active sensing machine is also capable of analyzing anything that crosses its path.

LIDAR is now used in automated and self-driving vehicles, robotics, surveillance, and agriculture.

Pressure Sensors

A pressure sensor is a device that senses pressure and converts it into an electrical signal.

Pressure sensors are used to measure the pressure of a gas or liquid by converting the physical pressure into an electrical signal. They are also good at measuring other variables such as speed and height.

Barometers and pressure gauges are the most popular pressure sensors used in IoT systems. Barometers help in weather forecasts, as they accurately measure the air nearby. Pressure gauges are used in industrial buildings, where they monitor the pressure in closed environments.

Pressure sensors can be used in very different areas. For example, they are employed in touch screens and biological medical instrumentation, and they are used in the industry, for example, in the automotive industry.

Water Quality Sensors

Water quality sensors are used to estimate water quality and to monitor ions, primarily in water distribution systems.

Water is used practically everywhere. These sensors therefore play an important role as they monitor the quality of water used for various purposes. They are in use in several industries.

Optical Sensors

Fiber optic sensor technology is used to detect electromagnetic energy such as light, electricity, and similar elemental particles. They can send, receive, and convert light energy into electrical signals.

Fiber optic sensors are used in energy, healthcare, aviation, chemicals, and environmental IoT platforms. Optical sensors can be ideal for environments such as oil refining, mining operations, pharmaceutical production, and chemical treatment.

We can expect high growth of fiber optic sensors as part of the increase in industrial applications in automation, as they are considered very suitable sensors for the IoT.

Chemical Sensors

Chemical sensors are used in several different industries. The goal of these sensors is to indicate changes in fluid or in the air. They play an important role in large cities, where it is necessary to record chemical changes to protect the population.

An important use of chemical sensors is found in industrial environmental monitoring and process control, detection of intentionally or accidentally released harmful chemicals, detection of explosive and radioactive materials, space station recycling processes, pharmaceutical industries and laboratories, etc.

Level Sensors

A sensor used to determine the level or amount of liquids or other substances flowing in an open or closed system is called a level sensor.

Level sensors measure the level of fluids. Level sensors can be used for smart waste management and recycling purposes. Other applications include measurement of tank levels, diesel meters, high- or low-level alarms, and irrigation control.

Level sensors are also commonly used in fuel gauges and fluid-level detectors in open or closed containers, sea and tsunami monitoring, water reservoirs, medical equipment, compressors, hydraulic reservoirs, machine tools, beverage and pharmaceutical treatment, high- or low-level detection, etc.

Infrared Sensors

An infrared sensor is an electronic sensor that measures infrared (IR) light radiating from an object nearby.

Infrared light has several applications. It can help doctors monitor the blood flow in humans, visualize heat leakage in houses, and identify environmental chemicals in the environment.

IR sensors are now used in a variety of IoT projects, especially in healthcare systems as they facilitate blood flow and blood pressure monitoring. They are even used in several common smart devices, such as smartwatches and smartphones. Other common applications include household appliances and remote control, breathing analysis, IR visualization (i.e., visualizing heat leakage in electronics, monitoring blood flow, looking below the surface of paintings), usable electronics, optical communication, non-contact temperature measurements, and blind angle detection.

IR sensors will play an important role in the smart-home industry, as they have a wide range of applications.

Image Sensors

Image sensors are devices used to convert images into electronic signals for display or storage to file.

The large use of image sensors is found in digital cameras, medical imaging and night vision equipment, thermal imaging equipment, radar, and sonar.

One of the most well-known uses includes the automotive industry, in which images play a crucial role. With these sensors, an automobile can recognize signs, obstacles, and many other things that a driver would generally notice on the road. They play a pivotal role in the IoT industry, as they directly affect the progress of self-driven cars.

They are also used in security systems, in which images help to capture details of a perpetrator.

In the retail industry, these sensors serve to collect customer data and help businesses gain a better insight into who is visiting their stores.

3.4 SENSOR FUSION

In the previous sections, we have seen that we can obtain information from many different sensors. If we get information about an event from more than one sensor, it may be an advantage to combine the different pieces of information.

Sensor fusion is the combining of sensory data derived from disparate sensors such that the resulting information has less uncertainty than when these sources were used individually.

Sensor fusion will reveal more about the context than a single sensor can provide. This is important in the IoT space, since a single thermal sensor, for example, has no notion of what causes a rapid thermal change.

With time-correlated data from multiple sensors, processing can make better decisions based on more data.

There are different methods for sensor fusion. Some of these methods use the central limit theorem (CLT), Kalman filters, or Bayesian networks.

3.5 ACTUATORS

Actuators are transducers that work in the opposite direction of sensors. An actuator takes electrical signals and converts them into physical action. An example of an actuator is an electric motor that creates movement.

An actuator is a mechanism that is responsible for moving or controlling something. An actuator requires a control signal and an energy source. The energy source may be electric current, hydraulic fluid pressure, or pneumatic pressure. When the control signal is received, the actuator responds by converting the energy into mechanical motion.

Electric Actuators

A motor that converts electrical energy into mechanical energy powers an electric actuator. It is a clean and easily accessible form of actuator because it does not directly use oil or other fossil fuels.

Hydraulic Actuators

A hydraulic actuator consists of a cylinder or fluid motor that uses hydraulic power to facilitate mechanical operation. The mechanical movement produces an effect that can be a linear, rotating, or oscillating motion. As liquids are difficult to compress, a hydraulic actuator can exert great force.

Pneumatic Actuators

Pneumatics is the utilization of energy by means of compression and expansion of gases.

A pneumatic actuator converts energy generated by vacuum or compressed air into either a linear or a rotary motion. Pneumatic energy can react quickly at start and stop, as there is no need for a reserve power source for operation.

Pneumatic actuators can produce considerable forces from relatively small pressure changes.

Thermal or Magnetic Actuators

Actuators that can be activated by thermal or magnetic energy have been used in commercial applications. Thermal actuators are often compact, lightweight, and economical, and they have high power.

Mechanical Actuators

A mechanical actuator works by converting some kind of motion, such as rotational motion, into another type of motion, such as linear motion. An example is a gear that runs around driving a vehicle moving straight ahead.

The operation of mechanical actuators is based on combinations of structural components, such as gears and rails, or pulleys and chains.

3.6 USE OF ACTUATORS

Actuators are devices that convert some type of stored energy into motion. Embedded systems in electric motors can create movement. Electric motors convert electrical energy into mechanical energy so that they can perform environmental operations. This leads to the following definition:

- Types of equipment that can convert electrical energy into mechanical energy are called actuators

By using an actuator, you get the opportunity to perform many different tasks, including robot control, activities at home such as watering flowers, camera control, unmanned aircraft, and 3D writing control.

One task that electric motors are often used for is the generation of rotation around a fixed axis to drive wheels, pumps, belts, and robot arms, for example.

There are three types of engines that are commonly used, namely servomotors, DC motors, and stepper motors.

Servomotors

A servomotor is an actuator that enables precise control of position, speed, or acceleration. It consists of a motor that is connected to a sensor that provides feedback about the position. Servomotors are controlled by sending an electrical pulse that determines how large the movement is.

Servomotors are small and extremely energy efficient. These features enable them to be used to control remote- or radio-controlled toy cars, robots, and aircraft.

Servomotors are used in applications such as robotics, CNC machinery, or automated manufacturing. CNC (computer numerical control) is the automated control of machining tools (drills, boring tools, lathes) by means of a computer. Servomotors are also used in industrial applications, robotics, in-line manufacturing, pharmaceuticals, and food services.

DC Motors

A DC (direct current) motor converts electrical energy into mechanical energy. The DC motor is the most common actuator used in electronics projects.

DC motors are used in many contexts from toys to advanced robots. They are ideal motors to use when there is a need for continuous rotation, as well as to drive the wheels of an electric vehicle.

DC motors are cheap and easy to use. They also come in a large selection of sizes to accommodate different tasks.

DC motors convert electrical energy into mechanical energy. The speed of rotation can be adjusted by the size of the power supply. Low power supply provides low rotation, and high power supply provides rapid rotation.

Stepper Motors

Stepper motors are DC motors that move with fixed steps. Stepper motors share a full rotation in a series of equal steps. The motor will rotate one-step at a time.

Stepper motors rotate a certain angle such as 1.8 degrees. This means that each time it receives a power pulse; it will rotate 1.8 degrees. This allows Stepper engines to rotate quite accurately with a rotation error of less than 5%.

With a computerized control, you can therefore achieve very precise positioning and/or speed regulation.

Stepper motors therefore rotate in a different way than DC motors, which rotate continuously, based on the amount of power supplied.

Linear Actuators

A linear actuator is an actuator that creates movement in a straight line, as opposed to the circular motion of a conventional electric motor.

Linear actuators are used in machine tools, in industrial machines, in peripherals such as disk drives and printers, in valves and dampers, and in many other places where linear motion is required.

Relays

A relay is an electrically operated switch. Many relays use an electromagnet to operate a switch mechanically, but other operating principles are also used, such as solid-state relays.

The advantage of relays is that it takes a relatively small amount of power to operate the relay coil, but the relay itself can be used to control motors, heating elements, lamps, or AC (alternating current) circuits that can themselves draw much more electrical power.

Relays are used wherever it is necessary to control a high-power or high-voltage circuit with a low-power circuit, especially when galvanic isolation is desirable.

4 ARCHITECTURE OF EMBEDDED SYSTEMS

An embedded system is made up of three components, as shown in Fig. 4.1.

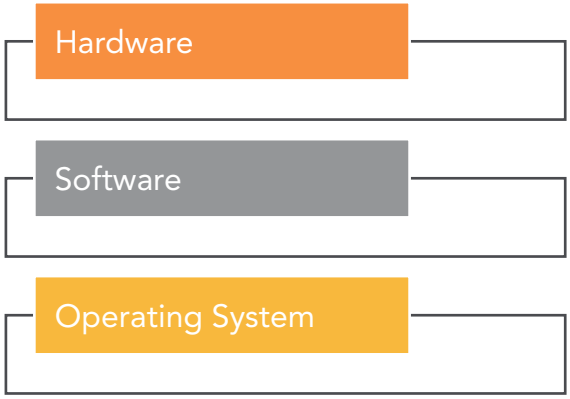


Figure 4.1 Main components of an embedded system.

We will look at the architecture of each of these components.

4.1 HARDWARE ARCHITECTURE

Embedded systems consist of electronic equipment placed on a circuit board. The main components of such a board are a processor, memory, data buses, and input and output devices.

Two different architectures for how these components work together are the von Neumann architecture and the Harvard architecture.

Von Neumann Architecture

The von Neumann architecture is a way of designing computers in which both program instructions and data reside in the same memory device. The processor is separate from the memory device.

The mathematician John von Neumann developed this type of architecture in 1945. He suggested an architecture consisting of the following components:

1. A memory that should contain both data and instructions in binary form
2. A processing unit that could perform mathematical and logical operations
3. A controller that interprets instructions in memory and ensures that they are executed
4. Input and output devices that provide communication between user and control unit

The von Neumann architecture supports simple hardware. It enables the use of a single memory. It also has a small memory (cache) near the processor.

The von Neumann architecture is used in personal computers, laptops, and workstations. All modern computers use this architecture.

Harvard Architecture

The Harvard architecture uses different memory devices for program instructions and data. Instructions and data also use different data buses. This enables the processor to access both instructions and data at the same time.

In a system with a pure von Neumann architecture, instructions and data are stored in the same memory, so instructions and data are retrieved over the same data bus. This means that a central processing unit (CPU) cannot read an instruction and perform data storage at the same time.

In a computer that uses the Harvard architecture, the CPU can both read an instruction and perform a data storage access simultaneously. Using the Harvard architecture will thus be faster than using the von Neumann architecture.

The Harvard architecture is used in digital signal processors and microcontrollers.

Typical of microcontrollers is that they have little software and memory for data, and they benefit from the Harvard architecture for fast processing while simultaneously accessing instructions and data.

Microprocessors and Microcontrollers

Embedded systems are based on microprocessors or microcontrollers. Both types are designed to perform calculations.

Microprocessors have a slightly simpler construction than microcontrollers, since the microprocessor consists only of one CPU and thus requires the connection of other components as well as memory chips. Microprocessors are used in various areas of technology. For example, they are present in mobile phones. They are also used in MP3 players, refrigerators, microwaves, some remote controls, printing devices, GPS receivers, etc.

Microcontrollers, on the other hand, are designed as independent devices. Microcontrollers have not only a CPU but also memory and external devices such as flash memory, RAM, or a serial communication port. The majority of microcontrollers in use today are embedded in other types of machinery, such as automobiles, robots, telephones, medical equipment, household appliances, and peripherals for computer systems.

ARM Processors

ARM is short for advanced RISC machine. RISC is an abbreviation for reduced instruction set computer. The ARM architecture is used in many products. The reason that the ARM processor has become a success is that it is small, is relatively inexpensive to produce, and has low power consumption.

ARM processors have low power consumption, making them well suited for use in portable devices. Almost all modern mobile phones and personal digital assistants contain ARM processors. This makes them the most widely used 32-bit microprocessor family in the world. ARM processors currently account for over 75% of all 32-bit embedded processors and are used in 98% of mobile phones sold every year.

In tablets, ARM processors provide longer battery life because they use less power. ARM processors also produce less heat than larger Intel processors, enabling tablets to be thinner. While not as fast as Intel PC or portable processors, ARM processors still provide reasonable speed, especially for mobile computing.

4.2 SOFTWARE ARCHITECTURE

An embedded system is usually designed to perform one specific task. Performing only one simple task does not require large resources. The software in an embedded system is therefore often designed for the following:

1. Little available memory
2. Low processor speed
3. Minimal power consumption

Different embedded systems can be designed for very different tasks. Therefore, there are several types of software architecture for embedded systems. We shall look at different types of software architecture.

Simple Control Loop

In this design, the program consists of only a single loop. The loop calls a function that performs some task or another. This design is therefore called either a simple control loop or just a control loop.

Interrupt-controlled System

Some embedded systems are controlled by interrupts. This means that a specific event will call software that performs a task in the system.

This type of system is used if events need a short time to be executed and if event handling is simple. Usually, these systems run a simple task in the main loop as well, but this task can wait a little bit when an unexpected event occurs.

Cooperative Multitasking

A multitasking system resembles the simple control loop system, except that the program system is designed to perform multiple tasks, and each task has its own run environment. The advantages and disadvantages of collaborative multitasking are the same as for the control loop, except that it is easier to add new software.

Pre-emptive Multitasking or Multithreading

This type of system is using context switching, which switches threads in the processor. This system has an operating system kernel. Since code may damage data in another task, the programs must be carefully designed and tested. Shared data access must be synchronized with some kind of synchronization mechanism.

Simple Operating System Kernel

A microcore is a small and simple operating system core. The usual functioning is that the operating system kernel allocates memory and switches different running threads in and out of the CPU. Processes in user mode implement key functions such as file systems, network interfaces, etc. In general, microkernels succeed when the context switching and the communication are fast but fail if they are slow.

Embedded Systems With a Large Operating System Core

In this case, a relatively large operating system core is adapted to an embedded system. This gives programmers an environment similar to a computer operating system such as Linux or Microsoft Windows, which is therefore very suitable for development. The downside is that it requires significantly more hardware resources. It is often more expensive, and the complexity of these cores can lead to less predictability and reliability.

Examples of embedded operating system core are Embedded Linux and Windows IoT. Despite the increased cost of hardware, this type of embedded system grows in popularity, especially on the more powerful embedded devices such as wireless routers and GPS navigation systems.

Additional Software Components

In addition to the core operating system, many embedded systems have additional components in the upper layer. These components consist of network protocols such as TCP/IP, HTTP, HTTPS, FTP, and CAN. They can also contain storage functions such as FAT and have flash memory systems.

If the embedded device has audio and video features, the current drivers will be present in the system. As far as the monolithic cores are concerned, many of these components are included.

4.3 OPERATING SYSTEM ARCHITECTURE

An operating system (OS) is system software that manages computer hardware and software resources and provides common services for computer programs. If you want to make an embedded system, it must have an operating system. Very simple embedded systems can do without an operating system, but it is rare for embedded systems not to have an operating system.

Often, embedded systems use operating systems designed specifically for embedded use. For example, all mobile phones use an operating system made for mobile phones. The operating system handles the user interface and all the basic functions of the phone.

An embedded operating system is designed to be efficient and reliable. Efficiency often comes at the expense of losing some functionality. An embedded operating system has fewer features than a standard computer operating system. The embedded operating system is often adapted to the embedded system. Often, many of the usual operating system components are removed, as they are not needed.

The hardware that runs an embedded operating system is often limited in terms of resources such as memory. The operating systems often have a limited task adapted to run a particular program that performs a particular operation. In order to take advantage of the processing power of the CPU, software developers often implement critical code they write into the operating system. This machine-efficient language can potentially result in better speed and performance at the expense of portability and maintenance. Embedded operating systems are most often written in a system programming language such as C.

An embedded operating system can either be an operating system designed specifically for the embedded device, or it can be one of the many operating systems adapted to run on top of an embedded system. Common embedded operating systems include Symbian, Windows Phone, Windows IoT, and Linux.

Embedded Operating Systems vs. Computer Operating Systems

An important difference between most embedded operating systems and computer operating systems is that the software of an embedded operating system is part of the operating system, often so that the entire software is only a single executable. Often, the system can run only a single program. Unlike PC operating systems, embedded operating systems are unable to load and execute various applications.

Since embedded operating systems often run only one application, hardware has little memory, and a slow CPU is typically used. Embedded operating systems are typically programmed in machine language to optimally benefit from the limited computing resources. This means that the operating system is adapted to the hardware for which it was designed, and this operating system will not be compatible with other hardware systems with other configurations.

Commercial Operating Systems

There are many operating systems on the market. These operating systems have both advantages and disadvantages compared with free operating systems.

There are many commercial real-time operating systems, and many are from well-established and reputable suppliers. However, buying one of these systems is something that should be carefully considered. The company's size, product quality, and use are important factors.

An important requirement is the possibility of technical support. When buying an operating system, both the buyer and the seller make a long-term commitment. One side of the relationship is the consideration of possible CPU migration in the future. A well-established provider of real-time operating systems can deliver new versions of the operating system, and their product is probably designed to simplify upgrades. Good documentation is important and can be expected from a commercial real-time operating system vendor.

One disadvantage of commercial operating systems is that, technically, each embedded system is different. CPU, memory, and external devices vary from device to device. Moreover, the operating system must fit the embedded system. Commercial operating systems also require licenses.

Free Operating Systems

Free operating systems are often easily downloadable real-time operating systems that are quite popular. Linux is not a completely free operating system because a supported version of Linux is not free. However, a supported and packaged version of Linux is something most embedded developers are likely to spend money on.

The advantage of free operating systems is that you do not have to pay anything for the operating system, nor will you have to do so later, as there are no license fees. Free operating systems often include the source code, which is useful for reference as the documentation may be limited and it may be difficult to get support later. It is also a requirement for the configuration and transfer to a new hardware environment.

A disadvantage of free operating systems is that implementing an operating system on an embedded device is a long-term commitment, so the issue of long-term support is important. For a free operating system, you cannot rely on long-term support.

4.4 MIDDLEWARE

Middleware is software that can be defined as any type of system software that is not part of the operating system kernel, the device drivers, or the user applications. However, although middleware is not part of the operating system, some operating systems can integrate middleware into the operating system.

In an embedded system, middleware is system software that is usually located either on device drivers or on top of the operating system, and it can sometimes be part of the operating system itself.

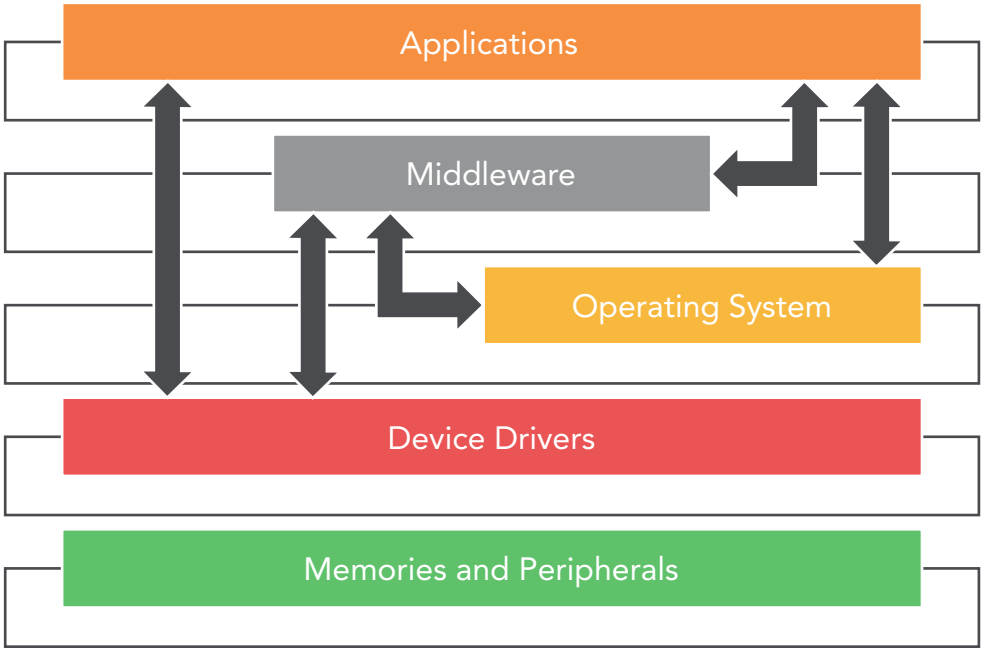


Figure 4.2 The embedded systems architecture.

Middleware is usually software that lies between applications and the core or driver software. Middleware can also be software that serves other software. More specifically, middleware is an abstraction layer commonly used on embedded systems with two or more applications to provide flexibility, security, portability, connectivity, interconnection, and application collaboration mechanisms.

An important advantage of using middleware is that it can reduce the complexity of the applications by centralizing the software infrastructure. However, using middleware in a system can affect scalability and performance. In short, middleware affects the embedded system in all layers.

There are many different types of middleware, but most types of middleware usually fall under one of the two following general categories:

- General-purpose middleware
- Market-specific middleware

General-purpose middleware is usually implemented in a variety of devices, such as network protocols, file systems, or virtual machines.

Market-specific middleware is unique to a particular family of embedded systems, such as a digital TV standard-based software that sits on an operating system or virtual machine.

4.5 SOME OPERATING SYSTEMS FOR EMBEDDED SYSTEMS

The following are some operating systems designed for embedded systems.

Linux

Linux can be used as an operating system in embedded systems. The benefits of using Linux as the basis for an embedded operating system include the following: supplier independence, low cost, open source, and hardware support.

Windows IoT

Windows 10 IoT Core is built for small, secure smart devices and supports ARM processors. With all the power of Windows, Windows 10 IoT shares all the benefits of developing Windows systems worldwide.

TinyOS

TinyOS is an embedded operating system and a platform for wireless devices that use low power. It is an open-source operating system, BSD licensed for low-power wireless devices. It is used in sensor networks, personal networks, smart buildings, and smart meters. The developer is TinyOS Alliance.

Contiki

Contiki is an operating system for network-based systems focusing on low energy used in wireless devices in the IoT. The open-source operating system is highly portable and supports multitasking for embedded systems in memory-efficient networks and in wireless sensor networks. Contiki is designed to run on types of hardware devices that are severely constrained in memory, power, processing power, and communication bandwidth. The developer is Adam Dunkels.

Mantis

Mantis is a multithreaded embedded wireless sensor network operating system. It is an open-source, multithreaded operating system written in C for wireless sensor network platforms. A simple C API enables Mantis operating systems to provide simplified programming of wireless sensor nodes.

Nano-RK

Nano-RK is a fully pre-emptive real-time operating system (RTOS) with network support for use in wireless sensor networks. Nano-RK supports fixed-priority multitasking to ensure that task times are met, along with support for the CPU, network, and sensors and actuators. Tasks can specify resource needs, and the operating system provides timely, guaranteed, and controlled access to CPU cycles and network packets. The developer is Carnegie Mellon University.

LiteOS

LiteOS is an open and interactive Unix-like operating system designed for wireless sensor networks. With the tools provided with LiteOS, you can operate one or more wireless sensor networks in a Unix-like manner, transfer data, install applications, retrieve results, or configure sensors. You can also develop applications for nodes and distribute such programs wirelessly to sensor nodes. LiteOS is open source, and the developer is Huawei Technologies Co., Ltd.

5 PROGRAMMING OF EMBEDDED SYSTEMS

The following sections describe some programming languages that are popular in the programming of embedded systems.

Many embedded systems are written in C or C++.

C is a good choice for embedded system development. C combines the low-level functionality of an assembly language very neatly with modern-day programming conventions. In C, porting embedded programs across different devices is much easier than in most other languages. It can be used on almost any existing advanced embedded system platform.

C++ is an object-oriented language based on C. If you want to develop slightly larger program systems, C++ may be preferable to C. The ability to use overloaded functions and constructors makes C++ an ideal choice for embedded systems programming. The object-oriented nature of C++ enables developers to program even the most complex embedded systems without overflowing the memory.

Embedded C++

Embedded C++ is a subset of the C++ programming language aimed at making embedded systems. The language includes only the parts of C++ that are used heavily in the embedded systems community and omits key C++ features such as exception handling, multiple inheritances, namespaces, templates, and virtual base classes.

Any standard C++ compiler can be used to compile embedded programs written in Embedded C++. Embedded C++ tries to avoid excessive memory consumption by removing most C++ core functionalities that are not exclusively used in embedded systems programming.

Python

Python has gradually become popular with embedded systems.

Python is in many ways a flexible language. What makes Python good for programming is its good readability. The design specifications for the language emphasize the importance of readable code and compact, elegant syntax.

Python may not be as useful for embedded programming as C or C ++, but with many available libraries, it is easy to implement functions. It is excellent for the automation of testing and data collection and analysis.

Java is widely used to develop embedded systems. Java is an object-oriented language that is highly portable.

Java makes it much easier to write extensible, portable, and downloadable embedded systems applications. A wide array of developer tools and powerful libraries make Java a suitable choice for embedded systems programming.

Go was developed by Google and is available for a variety of processors and platforms. Go is an open-source programming language that makes it easy to build simple, reliable, and efficient software.

Go comes with built-in features for unit testing, thus making testing your embedded application very easy. The rich API documentation of this embedded systems programming language is beneficial for both new and veteran developers alike.

Go adds an explicit hash table type, as well as types that can be very useful for collecting data from and sending data to separate sensors and actuators. The ability to process a network of sensors and devices is supported.

JavaScript

Programmers who make software for embedded systems are often familiar with scripting. They can often choose a scripting language because it is a fast way to solve problems.

JavaScript sounds like a variant of Java, but it is very different from Java. The two languages are similar in that there are, for example, some libraries that can be used by both, but the languages are developed separately and share no syntax or semantics.

The massive array of developer tools and third-party libraries makes JavaScript a suitable choice for developing fast and reliable embedded software. The event-driven, functional programming paradigm employed by JavaScript can be utilized to build stable embedded systems easily.

B# is a small, object-oriented programming language designed to run multiple threads. B# is designed specifically for small embedded systems. B# is similar to C#, but many of the features of C# that are not required for embedded projects are removed from B#.

B# was designed from the ground up as a small, highly efficient embedded control language. B# supports real-time control features. The embedded virtual machine allows B# to run on a variety of platforms. It uses only 24K memory, much less than what is needed for many of the other languages used.

C# is widely popular for building enterprise software. However, this innovative programming language is also used heavily for developing embedded systems for industrial purposes. With its strongly typed, component-oriented programming style, C# encompasses many useful features for embedded systems programming. Moreover, as Microsoft maintains this embedded systems programming language, you can easily find tons of documentation on this language.

Rust

Rust is one of the most modern programming languages being used to develop embedded systems. Rust combines the benefits of low-level languages such as C and C ++. From small microcontrollers to powerful single-board computers, Rust allows you to port your embedded system's code easily across a wide range of systems. Rust offers great community support.

Forth

Forth is a language designed and optimized for embedded system programming. It is a stack-oriented language and is primarily used for system-level programming. A language that has existed since the 1970s, Forth is still used today in many embedded systems (small computerized devices) because of its portability, efficient memory use, short development time, and fast execution speed.

Verilog

Verilog is an HDL (hardware description language) for developing electronic devices such as embedded systems. This is a widely used language in the field of embedded systems programming and offers very low-level access to system hardware. You can access and control almost every hardware-specific detail by incorporating this language into your embedded systems development.

Assembler

Machine code is the most basic code that can be used by the processor unit. The code is normally in hex code and provides basic instructions for each operation of the processor. This type of code is rarely used for embedded systems these days.

Writing in machine code is very laborious and time-consuming. It is difficult to understand, and it is difficult to search for errors in the code. To overcome this, high-level programming languages such as C, C ++, etc., are often used.

When you want to keep your project as compact as possible, Assembler is the language you want to choose. Assembler offers a way to pack and build clean machine code that is ultimately done by the processor. The advantage is that an expert can use optimizing tricks that are not available in any other programming language.

6 DESIGN OF EMBEDDED SYSTEMS

Every embedded system has an architecture. This is because an embedded system is composed of different components that work together, both software and hardware. An architecture consists of these components and the relationship between them.

If you want to design an embedded system, you must be familiar with the architecture of the equipment you want to create. Understanding the architecture of an embedded system is necessary for making a good system design. It is important to plan the design for embedded systems to avoid mistakes or an expensive result.

If you want to develop an embedded system, you must think about the following issues:

- The process of designing the system
- Equipment required
- System limitations with regard to processing power, memory, battery life, etc.
- The reliability and safety of the system
- The cost of the system
- Market and sale opportunities

All elements of an embedded system must interact with each other. Without understanding this interaction, it will be difficult to understand how the embedded system will behave under different conditions in the real world.

6.1 A MODEL FOR EMBEDDED SYSTEMS

All embedded systems are based on the model shown in Fig. 6.1.

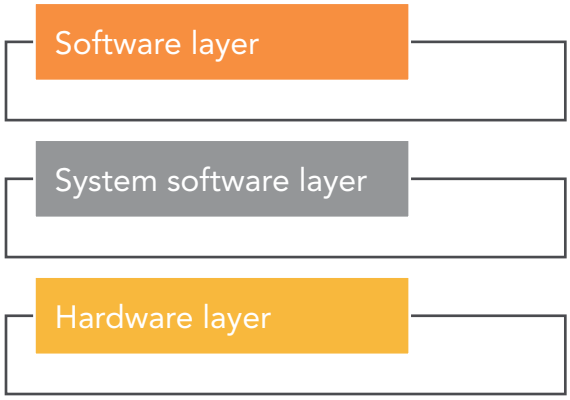


Figure 6.1 A model for the architecture of embedded systems.

This model is a layered model that represents the architecture of an embedded system.

Not all embedded systems have all these layers, but the hardware layer is always included. The hardware layer contains the physical components that are part of an embedded system.

6.2 STANDARDS OF EMBEDDED SYSTEMS

Some of the key components of embedded systems are made by specific procedures called standards. Standards dictate how components should be made and which other components the system needs to function satisfactorily.

The Institute of Electrical and Electronics Engineers (IEEE) is a professional association for electronic engineering and electrical engineering. The IEEE Standards Association (IEEE SA), a globally recognized standards-setting body within the IEEE, develops consensus standards through an open process that engages industry and brings together a broad stakeholder community. IEEE standards set specifications and best practices based on current scientific and technological knowledge. The IEEE SA has a portfolio of over 1,250 active standards and over 650 standards under development.

Standards can determine the functionality of all the three layers in the model for the architecture of an embedded system.

Standards can be classified as market-specific standards, general-purpose standards, or a combination of the two.

Most market-specific standards, except network and some TV standards, are often made for special groups of embedded systems. General-purpose standards, however, are often made for a market of embedded systems.

Programming language standards are an example of general-purpose standards. A programming language can be used in various embedded systems.

Network functionality standards can be implemented in all equipment that uses network communication.

Standards classified as market-specific standards define a functionality that belongs to a particular group of embedded systems.

Some examples of market-specific standards are the following.

Consumer Electronics

This group includes equipment used by consumers, such as PDAs, TVs, games, toys, home appliances such as microwave ovens, dishwashers, and washing machines, and Internet-enabled equipment.

Medical Equipment

Medical equipment is defined as instruments, apparatus, devices, and articles used alone or in conjunction with any other software included. These are equipment that can be used for diagnosis, prevention, reading, treatment, or control of the physical condition of patients.

Industrial Automation and Control

This group includes robotic equipment such as sensors, controllers of movement, human/machine communication equipment, and industrial switches.

Networking and Communication

This is equipment that connects endpoints in networks, routers, hubs, and switches. This category also includes equipment used for audio/video transmissions.

PART 2 THE INTERNET OF THINGS

7 WHAT IS THE INTERNET OF THINGS?

When embedded systems were connected to the Internet on a large scale, this led to something that we today call the Internet of Things (IoT).

The IoT means that any embedded device connects to the Internet so that it can communicate with other devices connected to the Internet without people being involved.

We can define the IoT as a network of connected devices in which each device has an IP address and embedded technology that enables it to communicate with other devices over the Internet. The IoT is a scenario in which objects, people, or animals are equipped with unique IP addresses so that they can transfer data over a network without people being involved in the communication.

The IoT is a gigantic network of connected things and people. It includes an extraordinary number of objects of all shapes and sizes: from smart microwave ovens that automatically cook at the right time, to self-driving cars that have sensors that detect obstructions in front of them, to portable exercise devices that measure your heart rate and the number of steps you have taken that day and then use this information to suggest training plans tailored to you.

A thing in the IoT can be something as different as a ceiling light, a human with a heart monitor, a refrigerator that alerts when goods run out, a jet engine filled with thousands of sensors that collect and send data, or a car that has built-in sensors to alert the driver when other cars or objects come too close.

The IoT consists of hardware devices that are equipped with electronics, software, sensors, actuators, and a network that enables the devices to communicate with each other. Almost all possible physical devices can be connected to the Internet.

What makes the IoT possible is the development of wireless technologies, microelectromechanical systems, and the Internet. Thanks to cheap processors and wireless networks, it is possible to make everything from shoes to airplanes part of the IoT. This enables devices that otherwise would be unintentional to get digital intelligence so that they can communicate with each other without people being involved.

The IoT will bridge the gap between the physical and digital world to improve the quality and productivity of people, communities, and industries.

How the Internet of Things Is Built

The following are the main components of the IoT:

- A thing
- A local network
- The Internet
- The cloud

A thing contains an embedded system that transmits and receives information over a network for the purpose of controlling another device or communicating with a user.

A thing in the IoT has one or more of the following components:

- A unique Internet address connection
- A communication device that can send and receive messages
- Built-in computer software that can perform information processing
- One or more sensors
- An actuator that can perform actions in the physical environment

In order for something to become a thing in the IoT, we must add some or all of the features mentioned above. This means that a chair, a refrigerator, or a lamp must contain an embedded system to become a thing in the IoT.

A key component of a thing in the IoT is a microcontroller or microprocessor that can execute software instructions.

Another key component is the IPv6 protocol that will have a central role in managing all the things that are and will be connected to the Internet. It is estimated that by 2020, 20 billion things will be connected to the Internet.

Machine-to-Machine Communication

The IoT is based on machine-to-machine (M2M) communication. M2M communication refers to direct communication between physical devices using any means of communication, including wired and wireless communication.

M2M communication and the IoT both deal with machines that communicate with each other, but there is a small difference between M2M communication and the IoT. M2M communication has traditionally been communication between two specific machines. The IoT, on the other hand, is equipment that communicates using an IP network and can thus communicate with any embedded device and computer connected to the Internet.

Common applications for M2M communication have been the traffic control system, telemedicine, company security, and telemetry.

M2M communication has been used in traffic control. In a typical traffic control system, there are sensors used to monitor the speed and size of traffic. This information is sent to computers that control the traffic. With such incoming data, M2M communication can regulate traffic flow.

8 EXAMPLES OF THE INTERNET OF THINGS

The IoT is used in many contexts, and its use only increases. In particular, IoT technology is used in smart homes, smart cities, connected cars, connected portable assets, and associated healthcare.

Market	Use
Smart homes	Heating, ventilation, air conditioning, lighting control, etc.
Wearables	Smartwatches that provide information about the user's health
Smart cities	Parking, health, pollution, traffic jams, lights, etc.
Connected cars	Operation, maintenance, brake control, finding parking spaces
IoT in agriculture	Soil moisture, fertilizer, control, etc.
Smart retail	Finding the best item placement for goods in a store
Energy connection	Discovering power outages faster
IoT in healthcare	Providing information about people's health
IoT in poultry and agriculture	Smart farming
Smart surroundings	Fire detection, air quality, earthquake detection

Figure 8.1 The table gives some examples of the use of the IoT.

Smart Homes

It will be nice to be able to turn on the air conditioning before you get home, to turn the light off automatically after you leave a room, or to unlock the door of your apartment to others for temporary access, even when you are not at home. The IoT can make life easier and more convenient for people.

Smart homes are currently popular. Embedded systems can be used extensively in the home, and it is in the home that most people probably encounter things connected to the Internet. This is a use of the IoT for which the major technology companies compete hard, especially Amazon, Google, and Apple.

Smart homes have become a success, and it is predicted that smart homes will soon be as common as mobile phones. The cost of owning a house can be a huge expense. The use of IoT technology in the home can save money, time, and energy.

Smart homes use more or less automated and intelligent features that will contribute to lower energy consumption, better comfort, easier operation, and a higher level of safety.

Smart homes often use different control systems for light, heat, refrigerator, fire protection, burglary protection, and ventilation. From your mobile phone, you can control light, heat, power consumption, blinds, garage door, exterior door, camera, ventilation, sound, and pictures in your home.

Smart homes can help older people live longer in their own homes instead of having to move to homes for the elderly. IoT equipment makes it easier for family and caregivers to communicate with them and monitor them.

Using the IoT as an aid provides a better understanding of how our homes work, and it enables you to save energy, for example, by cutting down on heating costs.

Example	Use
Light control system	You can turn on/off lights in your apartment from your mobile phone.
Heating	Smart thermostats can reduce monthly energy consumption by up to 30%.
Checking that the oven is switched off	Smart outlets can turn on/off any plugged device in your living room via the Internet.
Air quality	You can monitor the air quality of your home and the level of pollution in the city.
Monitoring an elderly family member	Wireless sensors are placed around the home so that you can follow a person's daily routine.
Monitoring a baby	Parents are provided with information about a baby's breath, skin temperature, body position, and activity level on their smartphone. Avoiding sudden infant death syndrome (SIDS).
Keeping your plants alive	Watering and grooming plants based on their actual growing needs and conditions saves time and resources.

Figure 8.2 The table gives some examples of the use of the IoT in homes.

Wearables

Wearables have become popular. Wearables in the form of activity gauges, sports watches, and smartwatches make it easy to track your health, chart how active you are, and set personal goals for your physical exercise. With an action camera mounted on your body, it is easy to film what you see wherever you are and what you experience.

Figure 8.3 Smartwatches have been designed to integrate themselves into every moment of the wearer's life, whether awake or asleep; they record heartbeats, sleep patterns, and workouts, among other aspects.

Wearables have experienced an explosive demand worldwide. Companies such as Apple, Google, and Samsung have invested heavily in building such devices.

Wearables are installed with sensors and software that collect data and information about the user. This data is later pre-processed to extract important information about the user.

The prerequisite for such IoT technologies to provide useful applications is that they are energy efficient and have a small size.

Smart Cities

Smart cities are an interesting application of the IoT. Smart monitoring, automated transport, smarter energy management systems, water distribution, city security, and environmental monitoring are all examples of the IoT used in smart cities.

By spreading many sensors over a city, the authorities will get a better idea of what is happening in real time. As a result, smart city projects are a key element of the IoT. Cities already generate large amounts of data from security cameras and environmental sensors and already contain large infrastructure networks used for some form of control, such as control of traffic lights.

Figure 8.4 Some uses of the IoT in smart cities.

The IoT will solve major problems for cities, such as pollution, traffic congestion, lack of energy supply, and so on. For example, sensors with mobile communication enable you to send alerts to municipal services when a trashcan can be emptied. By installing sensors and using web applications, citizens can find available parking spaces throughout the city. The sensors can also detect general errors or any type of problem in the city system.

Smart cities span several applications, from environmental monitoring to water distribution, waste management, traffic management, and city security. The popularity of smart cities is driven by the fact that many smart city solutions promise to reduce the problems of people living in cities. IoT solutions in smart cities reduce noise and pollution, solve traffic congestion problems, and help make cities safer.

Sensors can help the elderly in daily life, while others can keep track of whether a beach has become too crowded and then offer swimmers another option. Other examples are monitoring infrastructures such as roads, bridges, and railways with sensors to investigate structural changes such as cracks and tiles.

The ability to understand better how a city works should enable governments to make changes and monitor how these improve citizens' lives.

Example	Use
Smart parking	Monitoring of available parking space in a city
Structural health	Monitoring of vibrations and material fatigue in buildings, bridges, etc.
Noise pollution	Monitoring of noise in urban areas
Smartphone registration	Registering equipment that uses Wi-Fi or Bluetooth
Electromagnetic measurement	Measurement of radiation in homes or urban areas
Traffic jams	Monitoring of vehicles and pedestrians on the roads
Smart lights	Intelligent street lights that adapt to day and night and the weather
Garbage treatment	Keeping track of the amount of garbage in containers
Smart roads	Intelligent highways that warn about the weather and traffic conditions

Figure 8.5 The table gives some examples of the IoT in smart cities.

Industrial Internet

The Industrial Internet of Things (IIoT) is the new trend in the industrial sector. It is about industrial engineering with sensors, software, and data analysis to create intelligent machines.

The idea behind the IIoT is that smart machines are more accurate in processing data than humans. Moreover, this data can help companies detect inefficiencies and issues earlier.

The IIoT has great potential for quality control and data processing. Opportunities for stock information, tracking of goods, and automated delivery will increase the efficiency of the supply chain.

The following are some IIoT use cases and impacts:

- Preventive maintenance of new and pre-existing factory machinery
- Throughput increases through real-time demand
- Energy savings
- Safety systems such as thermal sensing, pressure sensing, and sensing of gas leaks
- Factory floor expert systems

Connected Cars

A connected car is a car equipped with Internet access and usually also with a wireless local area network. This enables the car to communicate with other devices both inside and outside the car via the Internet.

Connected cars are vehicles that use some form of communication technology to communicate with the driver, other cars, roadside infrastructure, or the cloud. This technology is able to improve not only traffic safety but also efficiency and comfort.

The car's digital technology has focused on optimizing the vehicle's internal functions. Now the focus is on improving the car's experience. A connected car is able to optimize its own operation and maintenance and to improve passenger comfort with the help of indoor sensors. Most large car manufacturers work with connected car solutions.

The IoT prevents accidents and improves vehicle safety. A vehicle can itself continuously monitor the near traffic and intervene if the driver is inattentive.

The Internet of Things in Agriculture

With the continuous increase in the world's population, the demand for food has become enormous. Governments help farmers use advanced techniques and research to increase food production. Smart farming is one of the fastest-growing fields in the IoT.

Farmers use information from IoT data to provide a better return on investment.

For outdoor farming, the IoT can measure soil moisture and pay attention to the weather, so that smart irrigation systems water only when needed, thereby reducing water consumption.

For indoor agriculture, the IoT offers the possibility of monitoring and controlling climate conditions such as humidity, temperature, light, etc. This will result in increased production.

Smart Sales

The potential of the IoT in the sale of goods is enormous. The IoT gives retailers an opportunity to connect with customers to improve the experience of the store.

Smartphones will be the way for retailers to remain connected to their consumers, even outside the store. Interaction via mobile phones can help resellers provide their consumers with better services. They can use IoT equipment to track customers' way through a store, improve the store environment, and place important goods in more trafficked areas.

You can use available sales data to identify which items are selling fastest and automatically adjust the sales data with supply so that popular items do not run out of stock.

The information provided by connected devices enables retailers to make smart decisions about which goods to fill, helping to save time and money.

Energy Transfer

The power grid of the future will not only be smart, but it will also be very reliable.

The basic idea behind smart grids is to collect data automatically and analyze the behavior of consumers and power suppliers to improve the efficiency and economy of electricity. You will also be able to detect the causes of power outages faster.

Energy Efficiency

People and organizations can achieve significant reductions in energy consumption by using the IoT. Sensors monitor lighting, temperature, energy consumption, etc. The data is processed by software to process real-time activities. For example, smart thermostats can automatically turn off heat/cooling to save energy when nobody is at home.

The Internet of Things in Healthcare

Health services have great opportunities to use the IoT. The concept of connected healthcare and smart medical equipment has enormous potential not only for businesses but also for increasing the well-being of people in general.

Research shows that the IoT in health services will be massive in the coming years. The IoT in the healthcare system aims to make people live a healthier life by using connected devices. The collected data will help the personal analysis of a person's health and provide tailored strategies to combat diseases.

The IoT allows for increased monitoring and recognition, which can help improve health and make life safer for people.

The Internet of Things in Poultry and Agriculture

Aquaculture monitoring is about animal husbandry and cost savings. By using IoT equipment to collect health and well-being data for cattle, farmers can become aware of sick animals early, which can help prevent many diseases.

Due to the size of the farming business and the large number of livestock that can be monitored, the IoT can revolutionize the way farmers work.

Smart farming becomes an important area of application in the IoT of countries that predominantly export agricultural products.

Disaster Warning

Sensors can collect critical information about the environment, which enables early detection of environmental disasters such as earthquakes, fires, tsunamis, etc. This will save human lives.

Law Enforcement

Better monitoring and investigation will enable the authorities to detect when a criminal act has occurred and respond much more quickly so that citizens can live more safely. Better law enforcement will even be able to predict crime and stop it before it happens.

Elderly

Monitoring of the elderly can save lives, as it automatically detects when someone falls or when someone gets a heart attack, so help can be sent immediately.

Environment Quality

Sensors can also detect radiation, pathogenic agents, and poor air quality so that hazardous concentrations can be identified early and people evacuated.

Smart Environment

The idea behind smart environments is to build an environment with embedded sensors and computing devices to better understand and control the environment.

Example	Use
Discovery of fires	Monitoring of forest areas
Air pollution	Control of carbon dioxide in factories and towns
Earthquake	Monitoring of earthquake-exposed areas
Water quality	Monitoring the quality of drinking water
Swimming pools	Checking the water quality in swimming pools
Pollution in the sea	Monitoring the level of pollution in the sea
Water leakage	Monitoring of humidity outside of water containers and water pipes
Floods	Monitoring of water levels in rivers, ponds, and reservoirs
Radiation levels	Monitoring of radiation at nuclear power plants
Gas leakage	Monitoring of gas levels and leakage in industry
Indoor air quality	Monitoring of toxic gases and oxygen levels
Temperature	Control of temperature in refrigeration disks in industry and medicine
Weather	Monitoring of weather conditions
Keeping the streets clean	Using real-time data collection to alert municipal services that a trash can needs to be emptied
More efficient street lighting	Smart lighting systems enable a city to provide the right level of lighting regardless of the time of day, season, and weather conditions

Figure 8.6 The table shows some examples of the IoT in smart environments.

9 ADVANTAGES AND DISADVANTAGES OF THE INTERNET OF THINGS

The IoT is beneficial for businesses as it enables the collection and analysis of data from production equipment, weather stations, smart meters, automobiles, and other types of machines.

IoT analysis programs can help companies understand data from sensors, with a view to reducing maintenance costs, avoiding equipment failures, and improving operation. In addition, consumer goods retailers, restaurants, and manufacturers can use data from smartphones, portable technologies, and home appliances to make marketing and campaigns more targeted.

To understand what impact the IoT can have on people's way of life, it is important to review the benefits and disadvantages of the IoT.

Advantage	
Automation and control	Automation is carrying out tasks without human intervention. When physical objects are connected and controlled with wireless infrastructure, there is a large degree of automation and control of the work. Automating tasks in a business helps increase the quality of services and reduce the need for human involvement.
Information	A person can access information from anywhere in the world. The more information, the easier it is for people to make good decisions.
Monitoring	The second most obvious benefit of the IoT is monitoring. Sensors provide information about the local environment. For example, sensors can tell when the refrigerator is empty or how good the air quality is in the house.
Economy	The economic aspect is the biggest advantage. IoT technology can replace people responsible for monitoring and maintenance. Optimal utilization of energy and resources can be achieved by using IoT technology and keeping the devices under surveillance. The IoT can alert you to any bottlenecks, breakdowns, and damage to the system. You will therefore save money by using IoT technology.
Saving time and being more efficient	The IoT helps people perform daily work tasks. This saves valuable time. Instead of doing monotonous tasks every day, it is possible for people to do other, more creative jobs. The amount of time saved by using the IoT can be quite large.
Better quality of life	All the uses of the IoT technology result in increased comfort, convenience, and better control, improving the quality of life for humans.

Figure 9.1 The table shows some benefits of the IoT.

Disadvantage	
Complexity	The IoT is a large and complex network that connects a huge number of different devices. A small error can affect the entire system, and the more complex the systems are, the greater the possibility of errors.
Less work for people	When tasks become automated, there will be less need for human labor. This will affect employment. In a future with the IoT, there will be a decline in the hiring of low-skilled employees.
Security	All IoT devices that a person uses are connected to the Internet. This creates a great risk of leakage of personal data. It is a major disadvantage of sharing information that confidential information is not safe and can be simply hacked by unauthorized persons.
Privacy	Privacy is a major issue with the IoT. All data must be encrypted so that information about your financial status or when there is no-one home in your apartment is not available to other people or criminals.

Figure 9.2 The table shows some disadvantages of the IoT.

The IoT makes our homes, offices, and vehicles smarter and more measurable and communicable. Although the IoT has some drawbacks, the benefits of saving consumers' time and money cannot be ignored. We must find ways to combat the disadvantages.

Security systems at home make it easier to monitor what is going on or to see and talk to visitors. Meanwhile, smart thermostats can help us warm up our homes before we get home, and smart light bulbs can make it look like we are at home even when we are out.

Sensors can also help us understand how noisy or polluted our environment can be. Autonomous cars and smart cities can change how we design and treat our local environment.

10 HOW THE CLOUD WORKS

10.1 WHAT IS THE CLOUD?

The cloud consists of remote servers that can be used via the Internet. Using the cloud is about storing and processing data on remote servers rather than on local and privately owned computers.

Figure 10.1 Cloud computing.

The cloud is the supply of data services over the Internet. Services you can use are servers, data storage, databases, networks, software, analysis, and more.

You can store small or large files in the cloud, and you can get hold of them everywhere via the Internet. You can use databases in the cloud. Computers in the cloud can process small or large amounts of data for you at any time.

The following are some common usages of the cloud:

- Saving, backing up, and restoring data
- Having websites and blogs
- Creating new apps and services
- Streaming audio and video
- Delivering software on request
- Analyzing data to draw conclusions

You probably use the cloud even if you are not aware of it. If you use an online service to send an email, edit documents, watch movies or TV, listen to music, play games, or save photos and files, it is likely that the cloud is making this possible.

Companies offering these computing services are called cloud providers, and they usually charge for their cloud services based on how large the usage is, just as you pay for the electricity you use in your home. You usually pay only for the cloud services you use, and this may reduce the operating costs you have. This makes the infrastructure more efficient, and the services are scaled, as the companies' needs change.

The first cloud services came into operation approximately ten years ago, but several organizations are already using this technology for various reasons.

The huge amount of data that IoT applications generate results in many companies choosing to use cloud computing instead of buying and using their own servers. The cloud giants already dominate these companies. Microsoft has its Azure IoT package, Amazon Web Services offers a variety of IoT services, and Google has Google Cloud.

Features of the Cloud

According to the National Institute of Standards and Technology (NIST), these are some specific features that define the cloud:

- Self-service on demand
- Wide network access
- Fast elasticity or expansion
- Pay as you go
- Measured service

Cloud computing resources can be provisioned without human interaction from the service provider. In other words, a cloud user can provision additional computing resources as needed without contact with the cloud service provider. This can be storage space, virtual machine instances, database instances, and so on.

Cloud computing resources are available over the network and can be accessed by diverse customer platforms. In other words, cloud services are available over a network, ideally high broadband communication links such as the Internet or, in the case of a private cloud, a local area network (LAN).

One of the great things about cloud computing is the ability to provision quickly resources in the cloud as businesses need them and then to remove them when they are no longer needed. Cloud computing resources can scale up or down rapidly and, in some cases, automatically, in response to business demands. It is a key feature of cloud computing. The usage and capacity, and therefore cost, can be scaled up or down with no additional contract or penalties.

In cloud computing, the user has to pay only for the service or the space they have utilized. The service is economical, and most of the time some space is allotted free. However, be aware that prices may suddenly rise in the future.

The Benefits of the Cloud

Without the cloud, the information technology (IT) growth and market would be non-existent. Essentially, billions of endpoint devices that were historically dumb and not connected would need to manage themselves without the ability to share data or aggregate data. The cloud provides the ability to have simple sensors, cameras, switches, beacons, and actuators communicate in a common language with each other. The cloud is the common denominator of the data currency.

The cloud is a big change from the traditional way companies have used IT resources. We will look at why the cloud is so popular.

Benefits	
Reduced cost	The cloud eliminates the cost of buying hardware and software, as well as setting up and running local data centers. It saves space and electricity around the clock for power and cooling, and you do not need your own IT experts to manage the infrastructure. The cloud saves businesses space, work, and money.
Saving time	Most cloud services are provided by self-service and on demand, so even large amounts of computing resources can be delivered in minutes, often with just a few mouse clicks. This gives companies great flexibility and removes the pressure on capacity planning.
Scaling	A major advantage of cloud services includes the scaling capability. In the language of the cloud, it means delivering the right amount of IT resources at the right time. For example, as much or as little computing power, storage space, and proper bandwidth can be provided as needed.
Productivity	Local data centers usually require a lot of equipment, hardware setup, software uploads, and various time-consuming IT management efforts. The cloud eliminates the need for many of these tasks, so IT teams can concentrate on other important tasks.
Performance	The largest cloud computing services run on a worldwide network of secure data centers that are regularly upgraded to the latest generation of fast and efficient hardware. These data centers provide better performance than a local business data center.
Reliability	The cloud makes data backup and disaster recovery simpler and more affordable. Data is stored more securely as copies can be located on servers in various locations around the world. If a server burns down or a data center goes out of operation, one can obtain the data from another data center.

Figure 10.2 The table shows common reasons why organizations use cloud services.

Disadvantages of the Cloud

There are also some disadvantages of using the cloud that should be taken into consideration.

Disadvantages	
Network connection dependency	You need a network in order to send files to the cloud and retrieve them. If you lose your network connection because of a storm or an outage, you may experience some downtime.
Limited features	Not all cloud providers are created equally. When you use cloud computing for storage and backup, you should ideally be working with a provider that offers unlimited bandwidth. You may also experience limited storage space or accessibility.
Loss of control	Essentially, you trust another party to take care of your data. You are trusting that they will maintain their data centers and servers with the same care as you would, if not more. You must trust that your provider's data centers are compliant and secured both physically and online. Some people find the lack of in-house control of the server unnerving.
Security	Recent cloud hacking cases have shown that not all cloud providers are as secure as they claim to be. As a business, you cannot afford to have sensitive information about your company, or your clients fall victim to hackers. One of cloud computing's greatest disadvantages is that you do not always know which providers you can trust.
Technical issues	If you experience any technical issues, you have no choice but to call your host provider's technical support for help. You cannot fix your cloud computing problems in-house, and some providers do not offer around-the-clock technical support.

Figure 10.3 The table shows some disadvantages of cloud services.

10.2 THE CLOUD ARCHITECTURE

Cloud computing refers to the components and subcomponents required in a cloud. These components usually consist of the following:

1. A front-end platform, which can be a computer or a mobile device
2. Back-end platforms, which consist of servers and a network such as the Internet or an Intranet.

Combined, these components constitute the cloud architecture.

In a cloud architecture, all applications are controlled, managed, and operated by a server in the cloud. The computer system is copied and preserved remotely as part of the cloud configuration. A good cloud computing system can create virtually unlimited efficiency and opportunities.

Front End and Back End

It is useful to divide a cloud-based system in front end and back end. The front end is the part near the user or client. The back end is the part of the system that is in the cloud. They are connected to each other via a network, usually the Internet.

The front end is the visible interface that computer users or mobile users use. The front end includes the client's computer and computer network, as well as applications required to access the cloud system. Not all cloud systems have the same user interface. Services such as web-based email programs utilize existing browsers such as Internet Explorer or Firefox. Other systems have different applications that provide network access for clients.

The back-end system includes all the resources required to provide cloud services. A system's back end can consist of a variety of servers, data storage facilities, virtual machines, a security mechanism, and services, all built in accordance with a distribution model, and all are responsible for providing a service.

The back-end system includes the various computers, servers, and data storage systems that are in the cloud. In theory, a cloud computing system can include virtually any type of computer program imaginable, from computing to video games. Usually, each application will have its own server.

Cloud Client Platforms

Cloud computing architectures consist of front-end platforms called clients or cloud clients. These clients are servers, fat clients, thin clients, tablets, and mobile devices. These client platforms interact with the cloud via intermediate software.

How the Cloud Works

Cloud computing resources are provided by server-based applications via digital networks or via the public Internet. The programs are available to users via mobile and desktop devices.

A central server manages the system and monitors traffic and the client to ensure everything runs smoothly. The server follows a set of rules called protocols and uses a special type of software called middleware. Middleware enables network-based computers to communicate with each other.

Cloud Storage

Network storage means data that is stored and available to several customers. Cloud storage is usually distributed in the following configurations: public cloud, private cloud, community cloud, or a combination of the three also known as hybrid cloud.

To be effective, cloud storage must be flexible, scalable, and secure.

If a cloud firm has many customers, this will lead to a high demand for storage space. Some companies have hundreds of digital storage devices. Cloud computing systems require at least twice as much storage space as required to keep all customer information stored. That is because these devices sometimes go out of operation, which can happen to any computer. A cloud system must have a copy of all customer information and store it on other devices. The copies enable the central server to access data that would otherwise be lost. Copying data as a backup is called redundancy.

Server Virtualization

It is possible to have several virtual servers on one physical server, with each virtual server running its own independent operating system. This technique is called server virtualization. By maximizing the performance of individual servers, server virtualization reduces the need for physical machines.

Virtualization uses multiple virtual machines on a physical computer or server. This achieves better scalability and workload while using fewer servers altogether. This configuration uses less power and saves money on infrastructure and maintenance.

In order to organize virtualization, a special operating system called a hypervisor is used. A hypervisor uses protocols that allow multiple virtual machines to run simultaneously on a physical server. The hypervisor controls the communication between its virtual machines and the connected world beyond.

Server virtualization used by hypervisors circumvents some of the physical limitations that stand-alone servers can face. Virtualization refers to the creation of a virtual machine that acts like a real computer with an operating system. This gives better utilization of hardware.

Type of hypervisor	Behavior
Native hypervisors	They run directly on a single metal server without an intermediate operating system and thus have full responsibility for performance and reliability.
Built-in hypervisor	They are assimilated into a processor on a separate chip and improve server performance.
Hosted hypervisors	These run as a clear software layer over both the hardware and the operating system. This type of hypervisor is beneficial for both private and public cloud computing to achieve performance improvements.

Figure 10.4 The table shows different types of hypervisors.

11 HOW THE INTERNET OF THINGS WORKS

The significance of the IoT lies in the interpretation of the IoT data and the decisions that are made on the basis of these data. The value of the IoT is what the data can tell us.

Analyzing data from IoT devices makes it possible to create systems that are more efficient. Smart IoT systems make it possible to automate many tasks. This applies in particular to tasks that repeat themselves or are time-consuming or dangerous.

The following aspects are what we usually want to achieve with an IoT system:

- Communication
- Control
- Cost savings

Communication

The task of an IoT device is to collect and communicate information. For example, a sensor can measure the temperature of a refrigerator and send a message to a cell phone if the temperature becomes too high or too low. Another example is an IoT device reporting whether an air filter is clean and working properly.

Control

You should be able to control an IoT device over the Internet, or it should be able to control itself. An example is a lamp that can be switched on and off using a mobile phone app. You can also use the IoT to start your washing machine from your mobile phone, and when the washing is finished, it can send a message to your phone.

Cost Savings

The desire to save money is the most common reason for businesses to adopt the IoT. While many companies will use the IoT to save money, individuals will use the IoT to automate home processes. However, sometimes individuals also want to save money. An example is to use the IoT to reduce heating costs in apartments or to reduce fuel costs while driving.

11.1 THE MAIN COMPONENTS OF THE INTERNET OF THINGS

The IoT refers to billions of physical devices around the world that are connected to the Internet. Thanks to cheap processors and wireless networks, it is possible to connect everything from lamps to aircraft to the IoT. This enables devices that would otherwise be unintentional to get digital intelligence so that they can communicate with each other without people being involved.

An IoT system consists of several different components. A thing is a central component of an IoT solution. A thing collects information, which is used to get insights. These insights will be used to make decisions that result in some kind of action.

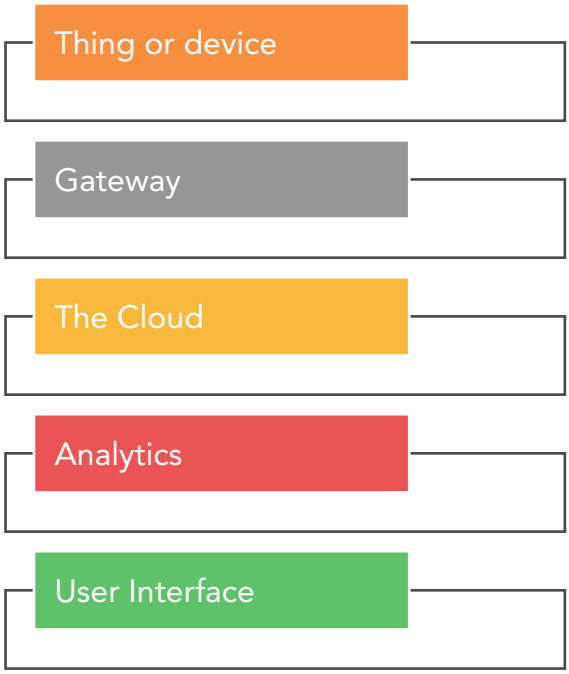


Figure 11.1 Major components of the IoT.

A thing is an embedded computing device or an embedded system that transmits and receives information over a network for the purpose of controlling another device or interacting with a user. A thing always has either a microcontroller- or a microprocessor-based device.

A thing can be a car or a washing machine, or it can be a larger system such as a jet engine filled with thousands of sensors that collect and transmit data. Smart city projects can fill entire regions with sensors to help us understand and control the environment.

Almost every physical object can be transformed into an IoT device if you can attach an embedded system to it and connect it to the Internet. However, the concept of the IoT is mainly used for devices that are not normally expected to have an Internet connection but can communicate with the network regardless of human action. For this reason, a PC is not considered an IoT device.

11.2 A MODEL FOR THE INTERNET OF THINGS

A complete IoT system has five important elements:

- Sensors or units with sensors
- Actuators
- Connection to a network
- Data management
- A user interface

Sensors or Units With Sensors

The IoT is often used to process measured data. Getting data from IoT devices anywhere in the world, connected via the Internet, provides the basis for control and optimization. This is what forms the basis for smart-city and smart-grid design.

First, sensors collect data from the environment in which it is located. This can be as simple as a temperature reading or something more complicated as a video recording. The expression units with sensors are used here because a device can have several sensors. However, if the thing is a stand-alone sensor or a unit with sensors, some kind of data from the local environment is captured in the first step.

Sensors are a key element in the connected networks for the IoT, which are increasingly being used in smart city and smart grid design. Connecting sensors to the IoT is relatively easy in some cases, but in other contexts requires more consideration. The data is sent as data packets to the IoT, usually using the IPv6 protocol. Digital interfaces for the latest sensors simplify the IoT interface to improve system reliability and functionality.

Connection

Data from sensors is sent to the cloud, but it needs a way to get there. Sensors and devices can be connected to the cloud using a variety of methods, such as mobile, satellite, Wi-Fi, and low-power wide-area network (LPWAN), or by connecting directly to the Internet via Ethernet.

The various options have advantages and disadvantages in terms of power consumption, range, and bandwidth. Selecting which connection option is best depends on the IoT application, but all the connection options perform the same task, namely getting data to the cloud.

Data Management

When the data reaches the cloud, the software performs some form of data processing. This can be very easy, for example, to check that the temperature reading is within an acceptable range. On the other hand, it can also be more complicated, such as using video software to identify objects such as an intruder in a house.

A user should decide what should happen if the temperature is too high or if there is an intruder in a house. Therefore, the system needs to send information to the user.

User Interface (Mobile Apps)

Mobile applications (also known as mobile apps) are software programs developed for mobile devices such as smartphones and tablets. The IoT is a network of internet-enabled devices all having an IP address and communicating with a user through a mobile app on a smartphone interface.

The information from IoT devices should be presented to an end user in one way or another. This can be via a notification to the user, for example, by email or SMS. For example, a text alert can be sent when the temperature in a refrigerator is too high. In addition, users can have an interface that enables them to examine the system. For example, a user can check the video footage in the house via a phone app or a web browser.

A user often has the opportunity to interact with the system. For example, the user can remotely adjust the temperature of a refrigerator via an app on a mobile phone.

In addition, some actions can be performed automatically. Instead of waiting for a user to adjust the temperature, the system can do it automatically by using a program. Moreover, instead of calling the user to warn about an intruder, the IoT system can automatically alert the police.

Summary

An IoT system consists of sensors that communicate with the cloud through some form of connection. When the data reach the cloud, the software processes them and may then decide to perform an action, such as sending an alert or automatically adjusting the system without notifying the user.

However, if user access is required or if the user wants to check the system, a user interface can be used. Any customizations or actions the user makes are then sent in the opposite direction through the system. A message from the user interface is sent to the cloud and then back to the sensor unit to make some kind of change.

12 BIG DATA AND THE INTERNET OF THINGS

The IoT can often produce huge amounts of data. Businesses can analyze these large amounts of data to gain knowledge of how components behave in real-life situations. This can help companies make improvements much faster. For example, data generated from sensors around a city can help city planners make the city community more efficient.

In the future, the IoT will generate larger and larger amounts of data. This will mean that companies need to upgrade their current storage systems, tools, and technology to be able to handle huge amounts of data and to take advantage of the insights that big data can provide.

12.1 BIG DATA

The term big data refers to data sets that are very large and complex. Special software that is designed to process large data sets is therefore often used. Large data tasks include data collection, data storage, data analysis, search, sharing, transfer, visualization, updating, querying, information, and privacy.

Big data often refers to the use of predictive analysis. Advanced data analysis methods and machine learning techniques are used to extract information from data. The analysis of the data sets can reveal relationships that can show business trends, prevent diseases, fight crime, and so on.

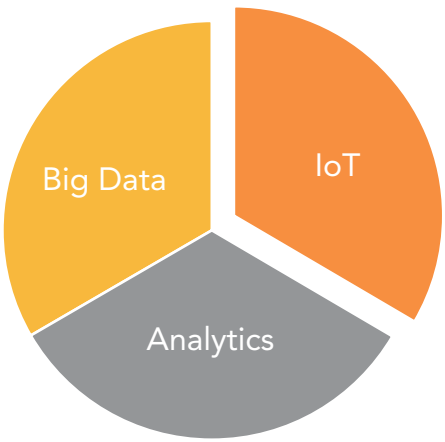


Figure 12.1 Big data and analytics are closely connected to the IoT.

Data sets only grow and grow. This is because they are increasingly being assembled by many and inexpensive sensors on the IoT, and they are transmitted via mobile devices, antennas, microphones, cameras, radio frequencies identification (RFID) readers, and wireless sensor networks.

Relational database systems, stationary statistics, and software packages to visualize data often have difficulty managing large data sets. The work may require massively parallel software running on a variety of servers. What counts as large data sets can vary depending on the user's capabilities and tools. For organizations facing hundreds of gigabytes of data for the first time, it may trigger a need to rethink data management options.

Data Storage

A consequence of the IoT is an increase in the volume of data that enters the data storage systems of businesses. There is a need to create new data centers to handle all this data.

Due to the enormous burden that IoT data will have on storage systems, organizations have begun to use a cloud-based solution, rather than using their own storage infrastructure. Unlike internal computer systems that need to be continuously updated as data load increases, cloud storage provides flexibility, scalability, regulatory compliance, and a suitable architecture for storing the IoT data.

Cloud storage options include public, private, and hybrid models. If a company has sensitive data that is subject to regulatory compliance requirements that demand increased security, it would be best to use a private cloud. For other companies, a public or hybrid cloud can be used for storing IoT data.

Businesses Need Big Data Technologies

Most companies need to improve their technologies to handle the vast amounts of IoT data that will come.

The most important task is to be able to receive data from IoT-linked devices. Devices can be connected to each other via Wi-Fi, Bluetooth, or other technology, and they must send messages using a well-defined protocol. One of the most commonly used protocols is MQTT, which is short for message queue telemetry transport. MQTT is a lightweight messaging protocol for small sensors and mobile devices, optimized for high-latency or unreliable networks. It is designed for connections with remote locations.

Once data is received, the next step is to find the best technology platform for storing IoT data. Many companies use Spectrum Scale, HDFS, or Lustre FS to store data. However, NoSQL databases such as Apache CouchDB are more suitable for IoT data, as they provide low latency and high throughput.

12.2 DATA COLLECTION

Data collection is the process of measuring values in the physical world and transforming these quantities into digital values that can be processed by a computer. Data collection systems, abbreviated as DAS or DAQ, convert analog waveforms into digital values for processing.

The components of data acquisition systems include sensors for converting physical sizes into electrical signals, as well as analog-to-digital converters to convert analog sensor signals into digital values.

There are also open-source software packages that provide all the necessary tools to retrieve data from IoT devices. These tools come from scientific environments in which complex experiments require fast, flexible, and customizable software.

12.3 DATA AGGREGATION

Data aggregation is any process by which information is gathered and expressed in a summary form, for purposes such as statistical analysis.

A common aggregation goal can be to get more information about specific groups of people based on specific variables such as occupation, age, or income. The information about such groups can then be used to customize websites with specific content and advertising that can appeal to a person belonging to one or more groups for which data is collected. For example, a site that sells music CDs might advertise certain CDs based on the age of the user and the data aggregate for their age group.

12.4 MACHINE LEARNING IN THE INTERNET OF THINGS

Machine learning is a subfield of computer science and is a type of artificial intelligence (AI) that provides machines with the ability to learn without explicit programming. Machine learning evolved from pattern recognition and computational learning theory.

In a very basic sense, machine learning in technology today is the process of elimination of human intervention wherever possible. It enables the data to learn patterns by itself and take autonomous decisions without a human having to write new code.

Devices connected to the IoT will create huge amounts of data, which will all be collected and stored. This data will be put into useable formats and silos by big data techniques. Machine learning will then use these huge oceans of data to improve processes and increase the self-sufficiency of systems. These processes are then fed back into the devices connected to the IoT, and the process can start again.

12.5 THE ROLE OF DATA ANALYSIS IN THE INTERNET OF THINGS

The value of an IoT system is not a single sensor event or a million sensor events archived away. The significance of the IoT lies in the interpretation of the IoT data and the decisions made on the basis of this data. The value of the IoT is what the data can tell us.

Data analytics (DA) is defined as a process used to examine large and small data sets. The difference between data analysis and data analytics is that data analytics is a broader term than data analysis. This is because data analytics also includes tools and techniques that data analysis does not use.

The purpose of data analytics is to draw meaningful conclusions from data sets. These conclusions are usually in the form of trends, patterns, and statistics that assist business organizations in effective decision-making.

Data analytics plays an important role in the growth and success of IoT applications and investments. The analytics tools enable business units to utilize their data sets effectively.

Key concepts in data analysis are volume and structure.

Volume

IoT applications use large amounts of data. Corporate organizations must handle and analyze these large amounts of data. These data sets, along with real-time data, can be analyzed easily and efficiently with data analysis programs.

Structure

Data sets from IoT applications can be characterized as unstructured, semi-structured, and structured data sets. There can be significant differences in the data formats and data types. Data analysis can analyze all these different sets of data using automated tools and software.

Types of Data Analysis

There are various types of data analysis that can be used on data from the IoT. Some of these types are streaming analytics, spatial analytics, time series analysis, and prescriptive analysis.

Streaming Analytics

This kind of data analysis processes and analyses large data sets, such as motion data sets. With this type of analytics, real-time data streams are analyzed to detect problems and immediate actions. IoT applications based on traffic analysis, financial transactions, aircraft fleet tracking, etc., can use this method.

Spatial Analytics

This is a data analysis method used to analyze geographical patterns to determine spatial relationships between physical objects. Location-based IoT applications, such as smart parking programs, can benefit from this kind of data analysis.

Time Series Analysis

As the name suggests, this kind of data analysis is based on time-based data that is analyzed to reveal related trends and patterns. Many IoT applications, such as weather forecasting programs and health monitoring systems, may benefit from this method of data analysis.

Prescriptive Analysis

This form of data analysis is a combination of descriptive and predictive analysis. It is used to determine the best practices that can be taken in a particular situation. Commercial IoT applications can take advantage of this kind of data analysis to draw better conclusions.

Use of Data Analysis in the IoT

There are scenarios in which IoT investments can benefit greatly from the use of data analyses. With the advancement in technology, new areas are emerging in which data analysis can be used in conjunction with the IoT. IoT analyses will also result in increased security and monitoring capabilities through video sensors and the use of data analysis methods.

Healthcare is one of the most important sectors in all countries, and the use of data analysis in IoT-based healthcare programs can make breakthroughs in this area. A reduction in healthcare costs, improvements in healthcare monitoring and remote healthcare, and better diagnosis and treatment can be achieved by using data analysis methods combined with the IoT.

The utilization of data analyses should therefore be promoted within the IoT field in order to obtain higher revenues, competitive profits, and customer engagement.

13 SOME BASIC TECHNOLOGIES IN THE INTERNET OF THINGS

The IoT will only grow as new IoT technologies are introduced. A problem with new technologies is that they require training to use them. To teach employees to use new technologies will be an important challenge for organizations that use the IoT.

There are many technologies related to the IoT, and many of them are new. In this chapter, we shall take look at some technologies that are important for the IoT.

IoT Platforms

IoT platforms combine in a single product many of the components of the infrastructure of an IoT system. The services from such platforms can be divided into three main categories:

1. Low-level control and operations on devices such as communication, monitoring, management, security, and firmware updates
2. IoT data collection, transformation, and management
3. IoT application development, including application programming, event-driven logic, visualization, analysis, and adapters to connect to enterprise systems

Enterprise systems (ES) are large-scale application software packages that support business processes, information flows, reporting, and data analytics in complex organizations.

IoT Device (Thing) Management

The things in the IoT need control and monitoring. This includes monitoring of devices, software updates, diagnostics, crash analysis and reporting, and physical and security management. There is a need for tools that are capable of managing and monitoring thousands and perhaps even millions of IoT devices.

IoT Analysis

We want to exploit the information collected by IoT devices in many ways, which will require new tools and algorithms for analysis. As data volumes increase over the next few years, analysis of IoT data will be different from traditional analysis.

Low-Power, Short-Range IoT Networks

Short-range and low-power networks will dominate wireless IoT connectivity in the near future, instead of connections using IoT networks over a wide area. Several different solutions will coexist, without any dominant wireless technology.

Low-Power, Wide-Area Networks

Traditional mobile networks do not provide a good solution for those IoT applications that need wide coverage combined with relatively low bandwidth, good battery life, high connectivity density, and low costs. Future standards such as narrowband IoT (NB-IoT) will probably dominate this need. NB-IoT is a low-power wide-area network developed by the 3rd Generation Partnership Project (3GPP) to enable a wide range of mobile devices and services. The 3GPP is a standards organization that develops protocols for mobile telephony.

IoT Processors

The processors and architectures used by IoT devices determine what capabilities they have, such as how good they are in terms of security, encryption, and power consumption, and whether they are sufficiently developed to support operating systems, up-to-date firmware, and embedded devices. Understanding the consequences of choosing a processor will require good technical knowledge.

IoT Operating Systems

Traditional operating systems such as Windows and iOS were not designed for IoT applications. They use too much power, need fast processors, and in some cases lack features that guarantee real-time responses. They also use too much memory for small devices, and they may not support the chips that IoT developers use. Therefore, a wide range of operating systems has been developed for the application of IoT devices to suit many different types of hardware and functional needs.

Event Stream Processing

Some IoT applications will generate extremely high data rates that need to be analyzed in real time. Systems that create tens of thousands of events per second are common, and even rates of millions of events per second can sometimes occur. In order to meet such requirements, distributed platforms have been developed that can process high-speed data streams and perform real-time analysis and pattern identification tasks.

IoT Standards

Standards and associated programming interfaces (APIs) will be crucial because IoT devices must interact and communicate, and many IoT business models will be based on sharing data between multiple entities and organizations. Many IoT systems will show up, and organizations that make IoT products may need to develop more variants to support different standards and systems; organizations should be prepared to update products during their lifetime when standards evolve and new standards and APIs arrive.

IoT Security

Security technology will be needed to protect IoT devices and platforms from hacking and physical manipulation. IoT security is complicated because many IoT devices use simple processors and scaled-down operating systems that may not support advanced security features.

Section 18, Internet of Things Security, will treat IoT security in more detail.

14 THE ARCHITECTURE OF THE INTERNET OF THINGS

The IoT is about connecting sensors and embedded devices to the Internet. However, for this to work, there needs to be an infrastructure to support the IoT units.

14.1 AN OVERVIEW OF THE INTERNET OF THINGS ARCHITECTURE

The IoT is a combination of several technologies. We shall look at the following important technologies:

- Sensors and sensor technology
- IoT gateways
- Cloud/server infrastructure and big data
- Mobile apps for end users
- IPv6 addresses

Sensors and Sensor Technology

A thing in the IoT is a device equipped with one or more sensors that collect data. These data are most often transmitted via a network. A thing may have one or more actuators that enable some kind of action, such as turning on or off a washing machine or starting or stopping a motor.

Sensors can provide many types of information, ranging from weather conditions and environmental conditions to motion on a conveyor belt, the health condition of a patient, or the maintenance data of a jet engine.

Sensors are everywhere and capture data from the environment. For example, a temperature sensor records the temperature in a room and transmits this information further via an IoT gateway.

IoT Gateways

An IoT gateway is an entry to the Internet for an IoT device. Gateways help bridge the local network of sensor units and the Internet or the World Wide Web. They do this by collecting data from sensor units and transferring them to the Internet infrastructure. An IoT gateway is the outward connection of an IoT device.

Data goes from the things to the cloud and back again through the gateway. A gateway that connects a thing to the cloud allows you to pre-process and filter data before moving it to the cloud. This reduces the volume of data sent to the cloud for processing and storage.

IoT gateways also receive commands coming from the cloud to the things. The things can then perform these commands using actuators.

Cloud Computing and Data Analysis

The data transmitted through a gateway is securely stored and processed in the cloud using a big data analysis engine. These processed data make our IoT devices smart with the ability to perform actions.

Data analysts can use the data on large storage servers in the cloud to look for patterns and gain insights. Data is analyzed and, in many cases, visualized with graphs or diagrams. For example, large data volumes can show the performance of devices, help identify inefficiencies, or create a way to improve an IoT system. That is, they can make systems more reliable and more customer oriented.

Mobile Phone Apps for Users

Mobile phone apps help users to control and monitor IoT devices. These apps provide information to a mobile phone and let users send commands back to their smart IoT devices.

Graphs, bars, and charts present information for users in an understandable way. We can also send commands from a mobile phone app to sensor units to change settings.

IPv6 Addresses

IPv6 addresses are the backbone of the entire IoT system. The Internet is concerned only with IP addresses and not with whether the address it refers to is a lamp or a refrigerator. IPv6 has as many as 3.4×10^{38} IP addresses. This will ensure that new IoT devices connected to the Internet in the future will be able to get a unique IP address.

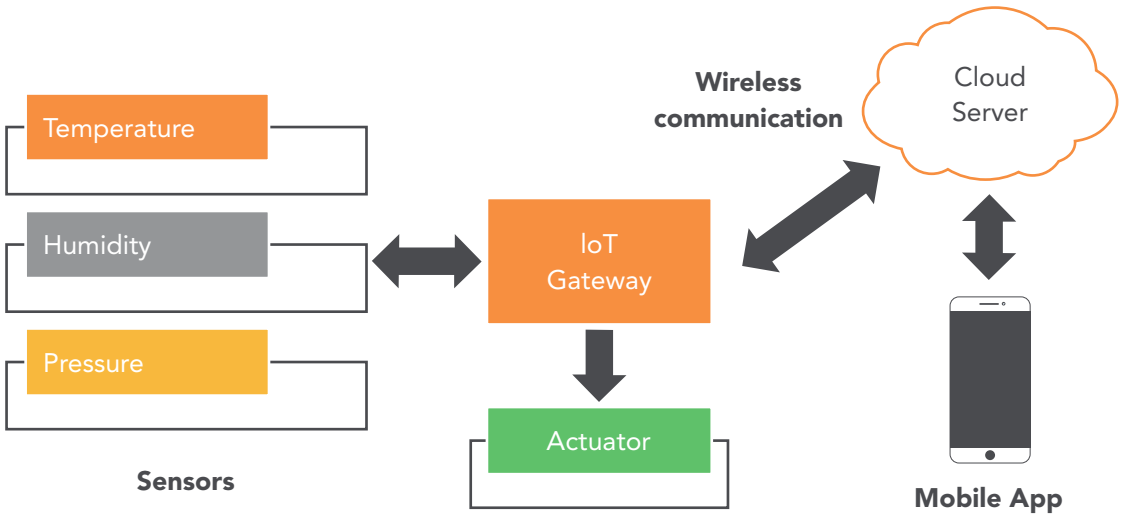


Figure 14.1 Figure of an IoT Architecture.

14.2 SOME REQUIREMENTS FOR AN IOT ARCHITECTURE

There are some requirements that are specific to IoT devices and the environments that support them. Some requirements come from equipment and software with specific limitations. Other requirements come from the manufacturing and use of IoT devices. The requirements are more about traditional consumer product design than existing Internet approaches. In addition, there are several existing best practices for server-side and Internet connections, which you must use.

The following are general requirements for some important parts of the IoT:

- Connection and communication
- Device management
- Data collection, analysis, and activation
- Scalability
- Security

Connection and Administration

Existing protocols such as HTTP are important for many IoT devices. However, HTTP and some other traditional Internet protocols can be a problem for two reasons. First, the memory size of the application may be a problem for small devices. However, the biggest problem is the requirement for low power consumption. To meet these requirements, IoT devices need a simple and small protocol.

In addition, there are IoT devices that connect directly and those that connect via gateways. The devices that connect to the cloud via a gateway often require two protocols: a protocol for connecting the device to the gateway and another one for connecting the gateway to the cloud.

Device Management

Many IoT devices are not properly managed, and this is often unfortunate. Active control of PCs, mobile phones, and other devices is becoming increasingly important, and the same path is both likely and desirable for IoT devices.

The following list provides some desirable requirements for managing IoT devices:

- The ability to update the software on a device
- Enabling or disabling certain hardware settings
- External remote configuration of Wi-Fi, GPRS, or network parameters
- The ability to disconnect a device that has been stolen
- Removing and securing data from a stolen device
- Tracking down a device that has been lost
- Updating security information

The list can be longer and can include features that may not be required or possible for some devices.

Data collection, Analysis, and Activation

What we want from an IoT system is that it can collect data from many devices. We want to save, analyze, and process the data.

An IoT architecture is designed to handle many devices. If these devices produce constant data streams, large amounts of data are created. This requires a highly scalable storage system, which can handle different amounts of data and large data volumes.

In some cases, reactions to incidents must be possible in almost real time, and this is a requirement for real-time analysis. In addition, the device must be able to analyze and process data. In some cases, this will only be simple computing, but for devices that are more complex, such as those dealing with incidents, we will need to utilize programs that are more powerful.

Scalability

A server architecture should preferably be highly scalable and be able to support millions of devices that constantly transmit, receive, and process data. However, a highly scalable architecture often has a high price, in hardware, software, and complexity.

An important requirement for this architecture is that it can support uses ranging from a few devices to many devices. Elastic scalability and the ability to distribute data in a cloud infrastructure are crucial. The ability to use small, inexpensive servers is a requirement for making a low-cost and large-format architecture.

Security

Security is important in connections with the IoT. IoT devices often collect personal data, and it is their job to bring real-world data to the Internet.

This leads to three categories of risk:

- Risks that are associated with any Internet system but that IoT designers may not be aware of
- Specific risks unique to IoT devices
- Security to protect against damage caused by, for example, abuse of actuators

The first category contains simple things like accessing IoT devices. An example is a refrigerator that is connected to the Internet and that has an unsecured SMTP server that sends spam.

The second category contains issues related specifically to IoT hardware, e.g. the fact that many units do not have enough security. For example, many IoT devices do not have resources to support proper asymmetric encryption. Another example is the possibility that someone may attack the hardware to understand security. These types of reverse engineering attacks are a problem compared with pure web solutions, in which there is often no available code to attack.

Two very important specific issues for IoT security are concerned with identity and access management. Identity is a problem with often poor practice.

14.3 IOT PLATFORMS

There is a need for an infrastructure to handle data streams from millions of devices attached to the IoT. The architecture of this type of real-time data stream processing needs to be able to handle data import, processing, storage, and analysis of hundreds of millions of events per hour.

An IoT platform (IoT framework) has three building blocks:

1. A thing
2. A gateway
3. Network and cloud computing IoT architecture

An IoT platform is a multilayer technology that enables straightforward provisioning, management, and automation of connected devices within the IoT system. It connects hardware, however diverse, to the cloud by using flexible connectivity options, enterprise-grade security mechanisms, and broad data processing powers.

IoT platforms originated in the form of IoT middleware, whose purpose was to function as a mediator between the hardware and application layers. Its primary tasks included data collection from the devices over different protocols and network topologies, remote device configuration and control, device management, and over-the-air firmware updates.

Many different structures for IoT platforms have been proposed to improve the connection of devices to the cloud. The key to successful IoT projects is collaboration between different units and solid IoT frameworks and platforms that enable powerful analysis. In addition, this platform must also offer value-creating services that make it attractive for companies to use the platform.

14.4 A FOUR-STAGE ARCHITECTURE OF AN IOT SYSTEM

A common IoT infrastructure is a four-stage architecture used in many IoT systems. We can look at these four stages as shares of a process. All four are built-in, mutually reinforcing architectures that carry data from the various networks (things) to traditional data centers for processing that provides users with information.

We can divide an IoT architecture into four stages.

Stage 1	Consists of sensors, actuators, and network services, often wireless.
Stage 2	Includes aggregation systems for sensor data and analog-to-digital data conversion. Aggregation means that data is processed and merged so that the amount of data becomes smaller.
Stage 3	Edge IT systems perform a pre-processing of the data before moving on to a data center in the cloud.
Stage 4	The data is analyzed, managed, and stored on data center systems in the cloud.

Figure 14.2 The table shows the four stages in an IoT architecture.

Stage 1. Sensors and Actuators

Sensors gather data nearby that can provide us with useful information. In a regular IoT system, a sensor can collect information and send it to a data center where it is processed, and in response to the data from the sensor, an actuator receives a command. Based on the data from sensors, actuators can act on the local environment.

The sensors provide the IoT data, and it is the information from the sensor data that makes the IoT system intelligent. Since data is central to the IoT, it is important to ensure that the data is accurate. Sensors and actuators may also have the task of ensuring the accuracy of the system.

In an IoT architecture, data processing can occur in each of the four stages of the infrastructure. However, even if you can process data near the sensor, there will be limited processing power available near an IoT device. Data is the central part of an IoT system, and you must choose whether the processing of data should be local at the sensor or in the cloud.

In order to get good information from the data, a comprehensive treatment is often required, and to get this you need to move the data to a data center in the cloud. However, not all decisions can wait until they are processed in the cloud. You need to deal with some decisions in real time. For example, did a robotic arm cut an artery when performing an operation in a hospital? Will a car crash? You do not have enough time to send such data to the cloud. You need to process such data close to the sensor on the edge of the network for the fastest possible feedback.

Stage 2. The Internet Gateway

The data from the sensors start in analog form. Further processing aggregates these data and converts them into digital data. Data acquisition systems (DAS) perform data aggregation and conversion. The Internet gateway receives the aggregated and digitized data and routes it via Wi-Fi, wired LAN, or Internet to systems in Stage 3 for further processing.

Systems in Stage 2 are often located near sensors and actuators. For example, a pump may contain half a dozen sensors and actuators that feed data into a data aggregation unit, which also digitizes the data. Usually, the IoT unit is physically connected to the pump. An adjacent gateway device or server will then process the data and pass it on to the systems in Stage 3 or Stage 4.

There will soon be large amounts of data from the analog data streams coming from sensors. Therefore, the data must be processed in advance. Values from the physical world that one might wish to measure are temperature, movement, tension, vibration, etc. This can create large amounts of data that are constantly changing. For example, an aircraft engine can generate huge amounts of data in a 24-hour period. Theoretically, there is no limit to the number of sensors that can feed data into an IoT system. In addition, an IoT system is always active. IoT data streams can be enormous; as much as 40 TB/second has been observed in one case. This is a lot of data to transport into a data center. It is therefore best to pre-process the data.

Another reason not to send the data to a data center is that analog data has specific timing and structural features that require specialized software to process. It is best first to convert the data into digital form, and that is what happens in Stage 2.

Intelligent gateways can build on basic gateway functionality by adding such features as analysis, malware protection, and data management services. These systems allow the analysis of data streams in real time.

Gateways are units at the end of the system (edge). DAS and gateway devices are used in a variety of environments, from the factory floor to mobile field stations, so these systems are usually designed to be portable, easy to deploy, and robust enough to withstand variations in temperature, humidity, dust, and vibration.

Stage 3. Edge Computing

Edge computing is data processing at the end of a network, that is, near the sensors or the data source. The purpose of performing aggregation and analysis at or near the data source is to reduce data traffic between sensors and central data centers.

Edge computing covers a wide range of technologies, including wireless sensor networks, mobile data collection, mobile signature analysis, networking, and processing. Other classifications of edge computing are local cloud computing, edge calculations, cloudlet, distributed data storage and retrieval, autonomous self-healing networks, remote cloud services, and more.

A cloudlet is a small data center or a cluster of computers designed to deliver cloud computing services quickly to mobile devices that are geographically close, such as smartphones, tablets, and portable devices. A distributed data store is a computer network in which information is stored on more than one node, often in a replicated fashion. An autonomous network runs with minimal to no human intervention and is able to configure, monitor, and maintain itself independently.

Once the IoT data have been digitized and aggregated, they are ready to be sent to a data center. However, the data may require further processing before going to a data center. This is where edge IT systems that perform more analysis come in. Edge IT processing systems can be located at external offices or elsewhere on the edge, but generally speaking, they are located at the facility or somewhere closer to the sensors, for example, in a switch cabinet.

Because IoT data can easily eat up network bandwidth and use up resources in data centers, it is best to have systems at the edge that can perform analyses that somehow reduce the burden on core IT infrastructure. If you send all the data directly to the data center, it would quickly require enormous capacity. You will also encounter security issues, storage issues, and delays. With an edge computing approach, you can pre-process the data, get meaningful results, and then forward them. For example, instead of transmitting raw vibration data for pumps, you can aggregate and convert the data, analyze them, and resend only projections when a device fails or needs service.

Stage 4. The Data Center and the Cloud

Data that needs a more thorough treatment but that does not need immediate feedback is forwarded to a physical data center or cloud-based systems, where more powerful IT systems can analyze, manage, and store the data safely.

It takes longer to get results when you wait for data at Stage 4, but you can do a more thorough analysis and combine your sensor data with data from other sources for deeper insight. Stage 4 processing can take place locally, in the cloud or in a hybrid cloud system, but the type of processing performed in this stage remains the same regardless of the platform.

14.5 IOT GATEWAYS

An IoT gateway is a gateway to the Internet for an IoT unit. An IoT gateway is a physical device or computer program that connects embedded devices with sensors to the cloud. All data moving to the cloud or the opposite way go through a gateway that can be either a hardware device or a computer program. An IoT gateway is also referred to as an intelligent gateway.

Some sensors generate tens of thousands of data per second. The edge provides a place to pre-process the data locally near the sensor before passing them on to the IoT gateway that transports the data to the cloud. The aggregation, summarization, and analysis of the data at the edge minimize the volume of data sent to the cloud, which can reduce response times and network transfer costs.

Another advantage of an IoT gateway is that it can provide additional security for the IoT network and the data it transports. Because the gateway manages information moving in both directions, it can secure data sent to the cloud and protect IoT devices from hacker attacks.

Gateways are an important part of the IoT communication structure, whose task is to connect sensor units to remote data centers in the cloud. Therefore, a gateway must have protocols that can communicate data over the Internet.

An IoT gateway creates a bridge between IoT devices, sensors, equipment, systems, and the cloud. IoT gateway devices offer local processing and storage solutions, as well as the ability to control sensor units automatically, based on the data input from the sensors.

14.5.1 HOW AN IOT GATEWAY WORKS

An IoT gateway performs multiple critical functions and has protocols for encryption, processing, managing, and filtering data.

IoT gateways also improve IoT systems in several ways.

Benefit	
Reducing network traffic	An IoT gateway can process data at the edge of the network so that only intelligent data is sent to the cloud. Data processing at the end means less traffic on the network.
Lower cost	Endpoint IoT units do not need high processing power, memory, or storage space since the gateway does all of this for them.
Faster systems	Faster and more advanced computing can significantly reduce time.
Minimizing risks	Gateways can isolate non-functioning devices and sensors before they cause major problems.

Figure 14.3 The table shows some benefits of using IoT gateways.

A Gateway Can Provide a Security Layer

As the number of embedded devices and sensors grows, communication across public and private networks is also increasing. Communication between things, the gateway, and the cloud must therefore be secure so that unauthorized persons do not manipulate or hack data.

Communication usually happens with a public key infrastructure (PKI) in which everything that communicates gets an identity, that is, a pair of cryptographic keys or a digital certificate that enables encryption of data. This can be difficult to handle without using an IoT gateway.

A Gateway Can Do Device Updates

Imagine you discover a security issue in one of your devices, or you notice that one of the sensors is too hot. Without a gateway, you need to make manual corrections because the devices and sensors have too little computing power to perform such tasks themselves.

If the system has a gateway, data about the sensors is sent to the gateway, and the gateway is configured to send firmware updates to all devices when necessary.

14.5.2 THE ARCHITECTURE OF THE IOT GATEWAYS

An IoT gateway is an important component of an IoT system, which should be efficient, secure, and easy to maintain.

The typical architecture of IoT solutions is usually complex. One of the most important factors that increase the complexity of IoT systems is the back-end services in the data center. IoT systems must handle a multitude of geographically dispersed devices. Because the properties of these devices are very different from those of web clients, desktop clients, and mobile clients, there is a need for an intermediate component that will act as a proxy between the sensor unit and the data center. What we need is an IoT gateway.

The Task of an IoT Gateway

We will describe the main reasons for introducing a gateway into an IoT architecture by discussing some of the key aspects of how the gateway architecture works.

First, sensors usually have very limited network connectivity capabilities. There is a need for a gateway that can provide sensors with a single point of contact with external networks using Wi-Fi, GSM, or another type of connection.

A gateway is not just an unintelligent proxy that transfers all the data from sensors to back-end services. Sending all the information collected by sensors to a data center will be very ineffective in terms of performance and network utilization. An IoT gateway is required to perform the pre-processing of information near the sensor unit before the data is sent to a data center. Such pre-treatment includes filtering and aggregation.

The gateway should also act as a place for local monitoring of a sensor unit area. You do not need to connect monitoring software to all sensors; it is easier to monitor only the gateway, which in turn is responsible for collecting all the necessary calculations from the sensors.

Overview of the Gateway Architecture

The following chart of a gateway architecture shows the most common architectural design of a gateway when it is not equipped with sensors. The gateway software installed on the device is responsible for collecting data from the sensor, pre-processing the data, and sending the results to the data center.

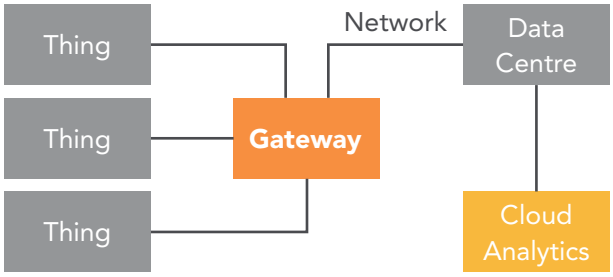


Figure 14.4 The connection between the things, the gateway, and the data center.

Summary

The gateway is a key component of all IoT solutions.

In choosing the right hardware for an IoT solution, it is very important to keep in mind that obtaining the right gateway software and management infrastructure is a factor that will have a major impact on the overall maintenance costs of the system.

14.6 EDGE COMPUTING

Edge computing is a way to divert data flow from IoT devices to the cloud. Before the data are transmitted from the sensors to the cloud, they are first pre-processed locally at the edge of the network, that is, near the sensor units.

In many situations, it can be beneficial to process data from IoT devices before sending them on a long journey to the data centers in the cloud. If computing is done before sending the data over the Internet to a data center, important data can be analyzed in real time. Many organizations have a need for this, for example, in industry, healthcare, finance, and telecommunications.

How Edge Computing Works

Edge computing is to process data locally near the sensor units. Then, the pre-processed data is sent for more processing and storage in a cloud data center.

One advantage of edge computing is that it reduces the amount of data sent to the cloud, thereby reducing traffic to data centers in the cloud.

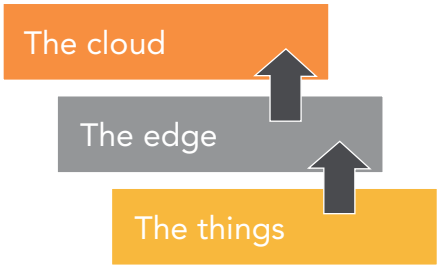


Figure 14.5 Edge computing enables data from the IoT to be analyzed at the edge before being sent to a data center or the cloud.

The IoT units transmit the data to a local unit, which performs data processing, storage, and networking. Data is first processed locally at the edge, and then all or part of the data is sent to the corporate data center or to a data center in the cloud.

Edge computing terminology	
Edge devices	An edge device can be any device that produces data. Edge devices can be sensors, industrial machines, or other devices that produce or collect data.
Edge	What the edge is may be different in different systems. In a company's IT system, the edge can be a computer. In a telecommunication field, the edge can be a mobile phone or a mobile tower. In an IoT car system, the edge of the network can be a car. In the industry, it may be a machine on a factory floor.
Edge gateway	An edge gateway is a transition between a device that performs edge computing and a larger network. Typically, an edge gateway will be a gateway to the Internet and the cloud.
Fat client	A fat client is software that can perform some computing. This is in contrast to a thin client, which is only able to transfer data.
Edge computing equipment	Various edge computing equipment such as devices, sensors, and machines will easily work in a computer system. You only need to make them available to the Internet.
Mobile edge computing	This refers to the development of edge computing systems in telecommunication systems, especially 5G scenarios.

Figure 14.6 The table gives a definition of some commonly used edge computing terms

Why Use Edge Computing?

Edge computing is ideal in a variety of circumstances, such as when the connection to the cloud is poor and it is not possible for IoT devices to connect constantly to a central cloud.

Other uses of edge computing are related to the real-time processing of information. Edge computing provides faster computing because data does not need to be sent over a network to a data center or to the cloud for processing. This is ideal in situations in which a waiting time of milliseconds can be unsustainable, which is the case in real-time applications.

The following provides an example of using edge computing. An oil rig in the sea can have thousands of sensors that produce large amounts of data, most of which have little significance. Perhaps the data only confirm that the systems are working properly. These data do not necessarily need to be sent over a network as soon as they are produced. By just sending important data over the network, edge computing reduces the data flow that goes through the network.

Security can also play a role in wanting to use edge computing. Some researchers argue that local security is theoretically better because data is not transmitted over a network and it stays closer to where it was produced. The less data there is in a corporate data center or a cloud environment, the less data is vulnerable.

Another use of edge computing is the construction of the next-generation 5G mobile network. Predictions say that when telecommunications providers build 5G in their wireless networks, they will increasingly add micro data centers that are either integrated into or located next to the 5G towers. Businesses will be able to own or rent space in these microcomputers to do edge computing and to have direct access to a gateway to the Telecom provider's wider network, which can connect to a data center in the cloud.

The vast amount of data from the world's mobile users is changing the planning of network infrastructure. Because of this, some researchers believe that edge computing itself is less secure because the edge devices themselves may be more vulnerable. Therefore, in the design of edge computing, security must be crucial. Data encryption and access control are important elements for protecting edge computing systems.

14.6.1 USE OF EDGE COMPUTING IN CARS

Edge computing is advantageous for autonomous cars. In autonomous cars, which are data centers on wheels, edge computing plays a dominant role. Autonomous cars will generate large amounts of data. Sending all this data to the cloud is uncertain, unnecessary, and impractical.

It is uncertain whether car data can be sent to the cloud because events on the edge must be processed in real time with very low latency. A self-propelled car that sends all data to the cloud as it drives on streets and highways could lead to a disaster. An example is a situation in which a person enters the street in front of a car. In this case, short waiting times for a reaction are required. The car must brake immediately. There is no time to send the data to the cloud for processing.

It is also unnecessary to send all the data to the cloud because most data have only short-term value. A person in front of a car is information that is usually uninteresting in retrospect. It is the reaction time that is important, not the data. Transporting large amounts of data generated from vehicles to the cloud is simply not practical.

However, the cloud still has a role in data from autonomous vehicles. How cars respond to an event can be valuable data because it provides information on how the car system tackles dangerous situations and what is happening in the traffic.

14.6.2 EDGE COMPUTING OR CLOUD COMPUTING?

Connected devices collect and process data from the physical environment to make life easier and better for us. Many companies use IoT data to understand their businesses better and to make good decisions. For example, a shipping company can place sensors on its pallets, crates, and containers to track geographic location, ambient temperature, pressure, and other environmental variables.

The IoT will only grow. With the rapid development of affordable sensors and IoT devices, many expect companies' use of IoT to explode. The consulting company Gartner predicts that companies will have installed approximately 7.5 billion connected things by 2020.

Experts expect this trend to generate around 44 trillion gigabytes of extra IoT data worldwide. This leads to an important question: What is the best architecture to handle all this data? There are three options for this:

1. Local architecture
2. Cloud-based architecture
3. Hybrid architecture

Local IoT Architecture

A local IoT architecture uses edge computing in which data is processed near the data source at the end of the network. It is estimated that by 2019, 45% of all IoT data will be stored and processed locally near the sensor units.

The edge computing architecture provides information that can help companies provide real-time data responses. For example, on an oilrig, sensors can detect whether a defective valve poses a fire hazard. In such a case, delays could have adverse consequences. If data is to be sent long ways to a data center and back before giving a warning to shut off the valve, it may be too late. Processing the data locally can reduce time delays and improve quick decisions.

Furthermore, a local architecture does not rely on Internet connections to data centers in the cloud. Moreover, companies seeking good data security favor local architectures. There are many uses in which local architecture and edge computing are beneficial.

Cloud Architecture

A cloud-based IoT architecture can be beneficial for organizations that have many connected devices where the desired information is based on a combination of internal and external data. For example, applications can benefit from the understanding of some data in relation to the aggregated view of all the data. Then only one set of data will lose its meaning outside a larger context.

In addition, cloud architectures offer greater opportunities to collaborate and interact with other IoT devices and cloud systems. This model provides far more architectural flexibility and influence of external data sources. The IoT distributions that utilize cloud architectures can be more effective because of the innovative and competitive offerings that only cloud providers can make available. Essentially, a cloud architecture can better secure IoT investments for organizations.

A Hybrid Architecture

Often, the best approach is one that effectively combines the local processing of large core data sets at the edge and then sends a reduced set of aggregated data for processing in a remote data center. As an example, smart cities that use car-parking sensors can process all sensor data near the garages and just send a summary of data about the number of vacancies in parking spaces to the cloud. This is not as expensive as sending all the data, and drivers who are going to park only need to know that there are vacancies in a garage. They do not necessarily want to know which exact parking spaces are available. In such cases, a hybrid architecture is ideal.

Another example is applications for wind turbines that use sensors to collect and analyze data on each wind turbine locally to optimize their performance. Here, many individual data points together provide a good insight into the health of the components of a turbine. Each component's health aggregated with the other related components provides an overview of a single turbine. Finally, aggregation of the summary from all turbines provides insightful and useful information about the wind park. In a situation like this, an important architectural assessment is how much data to treat front end and how much to treat back end. The combination of a real-time treatment of a local architecture with the cloud's system-based access and scalability provides the best of both worlds.

14.7 IOT MESH NETWORKING

Wireless mesh networks, an emerging technology, may turn the dream of a continuously connected world into reality.

What Is a Wireless Mesh Network?

A mesh network is a network of interlocked routers called nodes or points. These nodes work with one another to supply Internet coverage over a broad area. Each node spreads the radio signal a little farther than the last, minimizing the possibility of dead zones.

IoT devices often communicate with each other as in a mesh network and not always to a server in the cloud. An example is streetlights that send signals to each other.

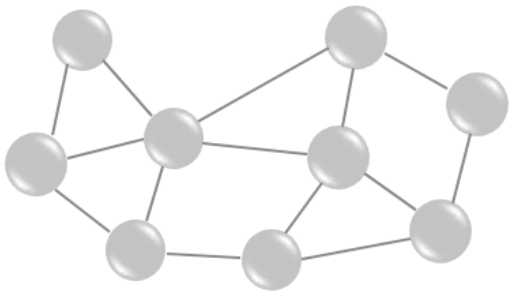


Figure 14.7 The figure shows the structure of a mesh network.

Why Use Mesh Networking for the IoT?

While wireless mesh networking technologies have been around for some time, only recently has the power of mesh reached a point of maturity alongside high availability from chip and silicon vendors. With newer, affordable costs, wireless mesh networking has become ideal for IoT builders.

Moreover, with the rise of connected homes and industry support of open-source resources such as Thread, mesh networking is now truly accessible while being sufficiently low cost to scale for production. As such, wireless mesh networking is becoming a much more viable, real choice for industrial and commercial IoT applications. It can provide additional services in systems where extending a connection between two nodes is limited.

Smart Cities

In a wireless mesh network, the network connection is spread out among dozens or even hundreds of wireless mesh nodes that talk to each other to share the network connection across a large area. Wireless mesh networks can easily, effectively, and wirelessly connect entire cities using inexpensive, existing technology.

Wireless mesh networking is great for extending radio signals through parking garages, campus grounds, business parks, and other outdoor facilities. Parking garages that utilize space availability checkers benefit greatly from mesh networks because they can extend the signal throughout the whole space and therefore can communicate when other clients have taken a spot.

Healthcare Equipment

Wireless mesh networks can help monitor and locate medical devices quickly. They can also act as a backup for medical equipment that always needs to remain online. If one node loses connectivity, another node can step in to keep the connection alive.

Smart Homes

Wireless mesh networks can help you track and manage temperatures across your house. You can set up one powered gateway and use temperature sensors and mesh-enabled nodes in each room to capture live data and adjust settings automatically.

Farming

Wireless mesh networking is also great for tracking sun exposure and water levels across crops. You can scale at low cost with mesh-enabled nodes across a whole acreage to create a cellular-connected IoT farm.

Industrial Internet

Wireless mesh networking can help you track pallets and monitoring large physical objects with a highly reliable wireless connectivity network. With wireless mesh networks, you can easily track key data across a factory floor and across multiple locations to identify issues before they happen.

Advantages of Mesh Networking

Mesh networking provides Internet in areas that do not have Ethernet connections or that are too far from the primary router.

A single point of failure is no problem, which is the issue in star topologies and bus topologies. If one node can no longer operate, the mesh network can reroute, which enables it to keep communicating between the remaining nodes.

Taking a mesh network down is impossible unless there is a worldwide catastrophe that wipes out all electronic devices in the world.

The mesh network works with minimal infrastructure and can therefore be deployed faster at a lower cost than traditional infrastructure.

Since the devices in a mesh network can retransmit signals farther, they have the ability to connect thousands of sensors over a wide area. A mesh network is also suitable for connecting devices in remote areas.

There is no centralized authority in a mesh network. Everything operating within the local network can run smoother because the nodes can communicate with each other instead of having to communicate with the central router.

Installation and management of most mesh networks are very easy because they are controlled with a companioning mobile app.

Extending the mesh network with more nodes is as easy as plugging the nodes into a power outlet and updating the app.

The setup might cost less than a traditional network if you consider the ease of adding nodes and the fact that very little installation needs to take place; you do not have to lay down any networking cables.

Disadvantages of Mesh Networking

Mesh networks are difficult to manage and troubleshoot. For big networks, one needs a strong mesh technology to make it worthwhile. This technology can be hard to find.

Battery life affects the availability of the nodes. For instance, a node with an empty battery could disrupt the network, causing more routing overhead and less reliability.

A mesh network system typically costs more than a traditional router. The cost of deployment can be problematic in certain scenarios. However, it can be redeemed by downloading a software development kit (SDK), which enables you to become a participant node in the whole mesh instead of building a mesh from scratch.

Market and regulatory forces make mesh networking difficult to deploy.

Mesh networks can replace Wi-Fi providers, phone carriers, and other intermediaries that provide connectivity to people. Consequently, intermediaries do not want to support this technology financially.

15 IOT COMMUNICATION

The Internet consists of many different technologies that communicate with equipment that can be anything from a single device to large platforms of embedded technologies and cloud systems connected in real time.

Connecting together components in the Internet requires protocols that enable devices and servers to communicate with each other. Many connections are required to link a widely branched and split IoT network.

Businesses want to take advantage of the benefits of the IoT, and this will cause the number of IoT applications to grow in the time to come. With a steady increase in suppliers that try to make homes, factories, vehicles, and health services more connected and thus smarter, it is important to understand the different standards that are in use.

In its simplest form, an IoT solution is a collection of sensors connected to a centralized management unit that allows the user to perform actions on the environment in one way or another. One example is to measure the temperature in a house and use this information to regulate the heating in the house. Another example is to monitor the operation of an assembly line and use this data to improve production.

Communication between IoT devices must use protocols of the same standard. Devices that connect together must be able to communicate with each other in order to transfer data. With different standards, the devices will have difficulty communicating with each other.

Compatible standards are therefore important for creating a large and reliable IoT network. Today, there are several IoT communication protocols and standards designed to simplify IoT design and increase the vendor's ability to innovate quickly.

15.1 SOME CENTRAL PROTOCOLS IN THE IOT

IoT devices must be able to communicate with each other. Data must be collected and sent to a server infrastructure. This server infrastructure must be able to store and share data and possibly send them back to IoT devices or to users.

A natural question is why there are any protocols outside HTTP to transport data across the WAN. HTTP has provided significant services and abilities for the Internet for over 20 years, yet it was designed and architected for general-purpose computing in client/server models. IoT devices can be very constrained, remote, and bandwidth limited. Therefore, more efficient, secure, and scalable protocols are necessary to manage the plethora of devices in various network topologies, such as mesh networks.

Today's Internet supports hundreds of protocols. The IoT will support hundreds more. It is important to understand what each of these protocols is designed for.

Some key protocols for IoT communication:

- MQTT, which is a protocol for collecting data from devices and communicating it to servers
- XMPP, which is a protocol that is good at connecting devices to humans
- DDS, which is a fast bus for the integration of intelligent machines
- AMQP, which is a queuing system designed to connect servers to each other

Each of these protocols is generally approved. There are at least ten implementations of each of them. All four protocols are real-time IoT protocols that can connect thousands of devices. Nevertheless, these protocols are very different.

15.2 SOME WIRELESS CONNECTIVITY TECHNOLOGIES

IoT devices can communicate with each other in many ways. Homes and offices often use standard Wi-Fi or Bluetooth Low Energy. Ethernet is also used if the devices are not particularly mobile. Other devices will use LTE or satellite connections to communicate. LTE (4G) is a standard for high-speed telecommunications. The large number of different communication options for the IoT has led to a need for better standards. Communication standards should be recognized and should work as well as Wi-Fi does today.

As the IoT evolves, it is likely that less data will be sent to the cloud for processing. To keep costs down, edge computing will probably be used. Data processing is then done locally near the unit and only aggregated data is sent to the cloud.

Both wireless and wired technologies are used in IoT communication. Wi-Fi 802.11 connection will continue to play a major role. It has good speed and low costs, making it attractive for many applications. It can be used over larger areas and requires one or more hotspots, which are easy to set up.

Figure 15.1 Some brands of wireless technologies.

Furthermore, ZigBee and other short-range wireless devices can be used in wireless sensor networks. Z-Wave can also be used in these networks. In addition, Bluetooth can create a personalized network (PAN) and thus adapt to specific applications. For example, Bluetooth Low Energy 4.0 and similar ultra-low energy versions are particularly suited for applications in medicine and physical exercise.

ZigBee and other 802.15.4 wireless device standards are very good options thanks to their low power consumption and ability to form wireless sensor networks. The 6LoWPAN Internet Engineering Task Force (IETF) standard enables you to connect billions of devices to the Internet by transmitting IPv6 packets with the 802.15.4 low power standard.

Two wireless technologies that are sure to be a success in the IoT are radio-frequency identification (RFID) and near-field communication (NFC). NFC, as well as RFID, uses the wireless area for communication. The area is limited to centimeters, but it is still useful for many applications. NFC-connected smartphones can replace credit cards to make payments: just press the vendor's NFC reader with your mobile phone to approve a purchase. Some smartphones already integrate NFC, but this is still in the early stages.

The IoT is a network of things with the possibility of communicating with each other. The most important aspect here is the connection between devices. How to connect the devices, and how should they communicate? How to develop wireless communication protocols?

We can connect the wireless communication protocols to the following six standards:

- Satellite
- Wi-Fi
- Radio frequency (RF)
- RFID
- Bluetooth
- NFC

Below, we will provide a brief overview of each of these communication techniques.

Satellite

Satellite communication makes it possible for a mobile phone to communicate to an antenna at distances of approximately 15 to 25 km. Depending on connection speed, this communication is called GSM, GPRS, CDMA, GPRS, 2G (GSM), 3G, 4G (LTE), or EDGE, among others.

Examples of satellite connectivity are measuring equipment that sends data to a remote server, or cars that are connected to the Internet.

An advantage of satellite communication is stable connectivity and universal compatibility. One disadvantage of satellite communication is that there is no direct communication from smartphone to device; the communication must go via a satellite. Satellite communication also has high monthly costs, high power consumption, and high response time.

Satellite is useful for types of communication that transmit small amounts of data. Up to now, it has primarily been used for industrial purposes, but in the near future, with the price of satellite communication gradually falling, the use of satellite technology can be much more viable and appropriate for consumers.

Radio Frequency

A radio-frequency (RF) signal refers to a wireless electromagnetic signal that is used as a form of communication for wireless electronics. Radio waves are a type of electromagnetic radiation, with radio frequencies ranging from 3 kHz to 300 GHz.

Radio-frequency communication is probably the simplest form of communication between devices. Protocols such as ZigBee or Z-Wave use a low-frequency RF radio, which is already built in, or later mounted onto, the electronic devices and systems.

Z-Wave has a range of approximately 30 meters. The radio-frequency band used is specific to each country.

ZigBee is based on the IEEE 802.15.4 standard, but its low power consumption limits transmission to a range of 10 to 100 meters.

An example of distance over which a radio-frequency connection works is the distance to your TV; the TV uses a radio-frequency signal that enables you to switch channels with a remote control. Other examples include wireless light switches, electrical meters, traffic control systems, and other consumer and industrial equipment that require short-range and low-speed wireless data transfer.

Radio-frequency identification (RFID) is a method for storing and retrieving data using small devices called RFID tags. An RFID tag is an integrated circuit that can be attached to or built into a product, animal, or person. RFID chips contain antennas that enable them to receive and respond to radio-frequency signals from an RFID transmitter. Passive chips respond with a weak radio signal and need no power source, whereas active chips transmit a more powerful response signal over slightly greater distances and require a power source.

Known applications for RFID are tracking, logistics, clothing, passports, ticketing, electronic payment, container terminals, product protection/burglar alarm, evacuation systems, access control systems, and animal identification.

RFID is a wireless use of electromagnetic fields to identify objects. Usually, you will install an active reader or read codes containing stored information, mostly authentication responses. Experts call it an ARPT (active reader passive tag) system. Short range for RFID is approximately 10 cm, while long range can go up to 200 m.

Advantages of RFID are that it does not require power and that it is an established and widely used technology. Disadvantages of RFID are that it is very uncertain, that it is running cost per card, and that it is not compatible with mobile phones.

Examples of use are data collection in a factory, animal identification, access duty, and access to buildings. RFID tags are also attached to a product so that the production and production process can be traced through the assembly line. For instance, medicines can be traced through department stores.

15.3 TYPES OF IOT COMMUNICATION

Central in the IoT system are the sensors, or rather the data from the sensors. The sensors can be placed anywhere in the world, and the data must be moved to a data center.

There are many technologies and data paths for moving data, and in this section, we shall give an overview of these technologies.

The IoT will combine personal area networks, local area networks, and long-range wide-area networks into a web of communication channels.

The technologies for communication of sensor data can be divided into the following categories

1. Non-IP-based wireless personal area network (WPAN)
2. IP-based WPAN and wireless local area network (WLAN)
3. Long-range communication systems

We separate out IP- and non-IP-based communication systems. Non-IP-based communication systems are optimized for cost and energy usage, whereas IP-based solutions, such as 802.111 Wi-Fi, usually have fewer constraints.

The different technologies for communication will be discussed below.

15.3.1 NON-IP-BASED WIRELESS PERSONAL AREA NETWORK (WPAN)

Sensors connected to the Internet need a method of transmitting and receiving information. This is the topic of the personal area network (PAN) and near-range communication. The communication to a sensor or an actuator can be a chopper wire or a wireless personal area network (WPAN).

WPANs are widely used. Wire-based connectivity is still used but primarily in industries and areas that are not radio-frequency friendly.

Some WPAN standards:

- 802.15 standards
- Bluetooth
- ZigBee
- Z-Wave

802.15 Standards

Many protocols and network models are based on or have a foundation in the IEEE 802.15 working groups. These working groups were initially formed to focus on wearable devices. Their work has expanded significantly and now focuses on higher data rate protocols and meter to kilometer ranges. Over one million devices with some form of IEEE 802.15.x protocol are shipped every day.

IEEE 802.15.4 standard forms the basis for many other protocols, including Thread and ZigBee.

Bluetooth

Bluetooth is a low-power wireless connectivity technology used widely in technology from cell phone sensors to keyboards and video games.

Bluetooth has been used extensively in IoT deployments for some time, being the principal device when used in low-energy mode for beacons (a beacon is a small Bluetooth radio transmitter), wireless sensors, asset tracking systems, remote controls, health monitors, and alarm systems.

ZigBee

ZigBee is targeted for commercial and residential IoT networking that is constrained by cost, power, and space.

Z-Wave

Z-Wave is a WPAN protocol used primarily for consumer and home automation and has about 2100 products using this technology.

Z-Wave's design focus is home and consumer lighting/automation. Z-Wave intends to use very low bandwidth for communication with sensors and switches.

15.3.2 IP-BASED WIRELESS PERSONAL AREA NETWORK (WPAN)

WPANs have adopted protocols that are typically not TCP/IP, at least from the outset. The protocol stacks for Bluetooth, ZigBee, and Z-Wave have similarities to a true TCP/IP protocol but do not inherently communicate over TCP/IP.

Some WPAN standards:

- WPAN with IP – 6LoWPAN
- WPAN with IP – Thread
- IEEE 802.11 protocols and WLAN

Internet Protocol and Transmission Control Protocol

Supporting an IP layer in a protocol stack does consume resources that could be applied elsewhere. However, there are key benefits in building an IoT system that allows devices to communicate over TCP/IP.

Regardless of the protocol used at the sensor level, the sensor data will ultimately be fed into a public, private, or hybrid cloud for analysis, control, and monitoring.

IP is the standard form of global communication for various reasons: ubiquity, longevity, scalability, reliability, and manageability.

WPAN With IP – 6LoWPAN

In an effort to bring IP addressability to the smallest and most resource-constrained devices, the concept of 6LoWPAN was formed in 2005.

6LoWPAN is an acronym that stands for IPv6 over low-power WPANs. The intent is for IP networking over low-power RF communication systems for devices that are power and space constrained and do not need high-bandwidth networking services.

6LoWPAN are mesh networks residing in the periphery of larger networks.

WPAN With IP – Thread

Thread is a relatively new networking protocol and is based on IPv6. Its principal target is home connectivity and home automation.

Based on the IEEE 802.15.4 protocol and 6LoWPAN, it has commonality with ZigBee and other IEEE 802.15.4 variants but with the significant difference that Thread is IP addressable.

Thread is also mesh based, making it attractive for home lighting systems. The philosophy of Thread is that by enabling IP addressability in the smallest of sensors and home automation systems, you can reduce power.

15.3.3 LONG-RANGE COMMUNICATION SYSTEMS AND PROTOCOLS

We shall look at some technologies for wide-area networks (WANs). Two WAN technologies are 4g-LTE and 5g, which are cellular, and there are some proprietary systems including long-range radio (LoRa) and Sigfox.

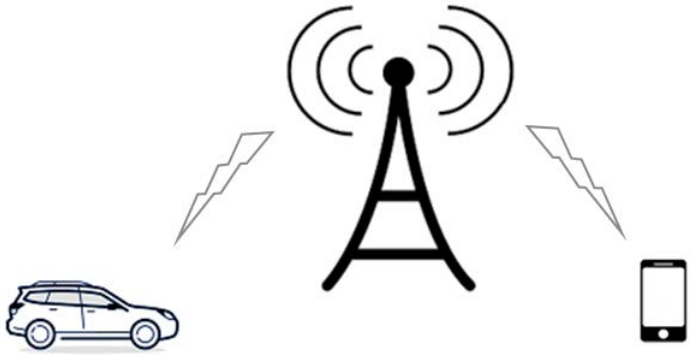


Figure 15.2 Long-range communication is usually a service, meaning that it has a subscription to a carrier providing cellular tower and infrastructure improvements.

4g-LTE and 5g

The most prevalent communication form is cellular radio communication. Although mobile communication devices existed for many years before the advent of cellular technology, they had limited coverage, occupied shared frequency space, and were essentially two-way radios.

The typical 4G-LTE network has three components: a client, a radio network, and a core network. The radio network represents front-end communication between the client and the core network and includes radio equipment such as the tower.

5G is the next generation IP-based communication standard designed to replace 4G-LTE. 5G promises to deliver substantial abilities for the IoT and commercial, mobile, and vehicular use cases. 5G also improves bandwidth, latency, density, and user cost.

LoRa and LoRaWAN

LoRa is a physical layer for a long-range and low-power IoT protocol, while LoRaWAN represents the MAC layer, which is responsible for moving data packets to and from one network interface card to another.

LoRa represents the physical layer of a LoRaWAN network. It manages the modulation, power, receiver, and transmission radios, as well as signal conditioning.

LoRaWAN is based on a star network topology. LoRaWAN relies on a cloud-based network interface.

Sigfox

Sigfox is a narrowband LPWAN protocol developed in 2009 in Toulouse, France. Originally, Sigfox was unidirectional and intended as a pure sensor network. That implies that only communication from sensor uplink was supported. Since then, a downlink channel has become available.

A Sigfox network can be as dense as one million nodes per base station. The density is a function of the number of messages sent by the network fabric. All nodes that attach to a base station will form a star network.

15.4 SOME COMMUNICATION PROTOCOLS USED IN THE IOT

Many communication technologies are well known, such as Wi-Fi, Bluetooth, ZigBee, and 2G, 3G, and 4G mobile phones, but there are also several new options. For example, Thread is an option for home applications, and white space TV technologies implemented in large cities are for wider IoT-based use. White space refers to unused frequencies in the wireless spectrum. Typical home Wi-Fi can go through only two walls, but white space broadband can go up to 10 kilometers through vegetation, buildings, and other obstacles.

Depending on the application, factors such as range, data requirements, security and power requirements, and battery life will determine the choice of technology combination.

Below, we will look at some important communication technologies offered to developers.

Message Queue Telemetry Transport (MQTT) is a protocol aimed at collecting data. As the name suggests, the main purpose is telemetry or remote monitoring. The goal is to collect data from many devices and transport the data to an IT infrastructure. The purpose of MQTT is that large networks with many small devices can be monitored or controlled from the cloud.

MQTT relies on the TCP protocol for data transmission. A variant, MQTT-SN, is used for other transports, such as UDP or Bluetooth.

MQTT makes it possible to monitor a large oil pipeline for leaks or vandalism, for example. Thousands of sensors are connected to a single main unit for analysis. When the system finds a problem, it can do something to solve the problem. Examples of the use of MQTT include monitoring of power consumption and lighting control.

The Extensible Messaging and Presence Protocol (XMPP) is an open technology for real-time communication that provides a wide range of applications, including instant messaging, presence, voice and video calls, collaboration, and generalized routing of XML data.

XMPP was designed to connect people via text messages. The name Extensible Messaging and Presence Protocol highlights the purpose of the use. Presence implies that people are closely involved.

XMPP uses the text format of XML, which is suitable for person-to-person communication. Like MQTT, it uses TCP or possibly HTTP over TCP. One of the strengths of XMPP is an address system of the form named@domain.com, which helps connect the devices in the large Internet system.

XMPP offers an easy way to add addresses to devices. XMPP is a good way to connect, for example, a thermostat at home to a web server so that you can access it from your mobile phone. The power of addressing, security, and scalability make XMPP ideal for consumer-oriented IoT applications.

Unlike MQTT and XMPP, Data Distribution Service (DDS) refers to devices that directly use data from other devices. It distributes data to other devices, and it is sometimes called middleware or a connectivity framework. DDS supports the user interface of the IT infrastructure, but the main purpose of DDS is to connect devices to other devices. DDS can effectively deliver millions of messages per second to many simultaneous receivers.

Devices have other computing requirements than IT infrastructures. First, devices must be fast. Real time is often measured in microseconds. Devices must communicate with many other devices in complex ways, so TCP's simple and reliable point-to-point currents are far too limited. Instead, DDS offers detailed quality control, multicasting, configurable reliability, and data copying. DDS offers powerful ways to filter and choose exactly which data goes where and where there may be thousands of simultaneous destinations. Some devices are small, so there are light versions of DDS running in limited environments.

High-performance integrated device systems use DDS. It is the only technology that provides the flexibility, reliability, and speed that are needed to build complex real-time applications. Applications include military systems, wind farms, hospital integration, medical imaging, real estate tracking systems, and car testing and security.

AMQP
Advanced Message Queuing Protocol

The Advanced Message Queuing Protocol (AMQP) is sometimes regarded as an IoT protocol. AMQP sends transaction messages between servers. As a message-centric middleware that has emerged from the banking sector, it can reliably process thousands of queue transactions.

AMQP is focused on not losing messages. It uses TCP for communication, which provides strictly reliable point-to-point connectivity. Furthermore, the end recipient must confirm receipt of each message. True to its origin in the banking sector, AMQP focuses its middleware on tracking all messages and ensuring that each message is delivered as desired, regardless of error or restart.

AMQP is mainly used in business communication. It usually defines devices as mobile phones that communicate with back-office data centers. In the IoT context, AMQP is most appropriate for server-based analysis functions.

Bluetooth is an important communication technology over short distances and has been given a very central role in computing and many consumer products. It is expected to be central to portable products that connect to the IoT but also in many cases via smartphones. The new Bluetooth Low-Energy (BLE) or Bluetooth Smart is an important protocol for IoT applications. It offers a similar range to Bluetooth but is designed to provide significantly lower power consumption.

Bluetooth is used in many products such as phones, tablets, media players, and robot systems. The technology is very useful for transferring information between two or more devices that are close to each other and in low-bandwidth situations. Bluetooth is often used to transfer audio data to phones with Bluetooth hearing clocks, or in file transfer for handheld computers.

However, Smart/BLE is not designed for file transfer but is more suitable for small data transfers. The Smart/BLE protocol has a great advantage over many competing technologies, due to its extensive integration into smartphones and many other mobile devices.

The advantage of Bluetooth is that it is an established and widely used technology. Bluetooth is incorporated in all smartphones, for which the technology is continually upgraded and improved through new hardware.

Disadvantages of Bluetooth are that hardware features change very quickly and need replacing, that the life of a battery-powered iBeacon is between 1 and 2 years, and that if people turn off Bluetooth, there are problems with its use.

Bluetooth technology is definitely one of the most interesting technologies right now, but its functionality is often overstated or misunderstood.

ZigBee is a high-level communication protocol used to create personal area networks with small, low-power digital radios, such as networks for home automation, medical device data collection, and other low-power low-bandwidth needs, designed for small-scale projects, which need wireless connection.

ZigBee is widely used in home automation and in industrial environments. ZigBee PRO and ZigBee Remote Control, which are some of the available ZigBee profiles, are based on the IEEE802.15.4 protocol, which is a wireless network technology for the industry. This protocol operates at low data rates over a limited area and within a 100-meter range, which suits a home or building. ZigBee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach devices that are more distant.

ZigBee has some significant benefits in complex systems. It offers low power consumption, high security, robustness, and high scalability and is well positioned to utilize wireless control and sensor networks in M2M and IoT applications. The latest version of ZigBee is 3.0, which is essentially the unification of the various ZigBee wireless standards into a single standard.

Z-Wave is a low-frequency RF (radio frequency) communication technology designed primarily for home automation and for products such as lighting control, security systems, thermostats, locks, and garage door openers, among many others. Optimized for reliable communication of small data packets with data rates of up to 100 Kbit/sec, it operates below the 1 GHz band and resists interference from Wi-Fi and other wireless technologies in the 2.4 GHz range such as Bluetooth or ZigBee. The range is approximately 30 meters.

It supports entire networks without the need for a coordinator node and is highly scalable, allowing up to 232 units to be controlled. Z-Wave uses a simpler protocol than most other communication technologies, thus supporting quick and easy development.

6LowPAN (IPv6 low-power wireless personal area network) is an important Internet protocol-based technology. The target for IP networking for low-power radio communication is applications that need wireless internet connectivity at lower data rates for devices with a very limited form factor. Examples are automation and entertainment applications in home, office, and factory environments.

The standard can be used on multiple communication platforms, including Ethernet, Wi-Fi, 802.15.4, and sub-1GHz ISM. A key attribute is IPv6 (Internet Protocol Version 6). IPv6 is the successor to IPv4 and offers approximately 5 x 1028 addresses for each person in the world, enabling any embedded object or device in the world to have its own unique IP address and connect to the Internet.

6LowPAN is designed to send IPv6 packets over IEEE802.15.4-based networks and to implement open IP standards including TCP, UDP, HTTP, COAP, MQTT, and web sockets, offering end-to-end addressing nodes standards that enable a router to associate the network with IP. 6LowPAN, which is a mesh network that is robust, scalable, and self-repairing.

Thread is a brand new IPv6 network protocol aimed at home use. Based on 6LowPAN, Thread is not an IoT application protocol like Bluetooth or ZigBee. However, from an application point of view, it is designed primarily as a complement to Wi-Fi. It is recognized that although Wi-Fi is good for many consumer devices, it has limitations in its use for the IoT.

Launched in 2014, this royalty-free protocol is based on various standards, including IEEE802.15.4, IPv6, and 6LoWPAN and offers a robust IP-based solution for the IoT. Thread supports a network using IEEE802.15.4 radio transmitter modules and can handle up to 250 nodes with high levels of authentication and encryption. A relatively simple software upgrade should allow users to run Thread on existing IEEE802.15.4-enabled devices.

Wi-Fi is a WLAN that uses the IEEE 802.11 standard through 2.4 GHz UHF (ultra-high frequency) and 5 GHz ISM (industrial, scientific and medical) frequencies. Wi-Fi gives Internet access to devices that are within a range of approximately 66 meters.

Wi-Fi connectivity is often an obvious choice for many developers, especially in terms of home Wi-Fi. Wi-Fi offers fast data transfer and the ability to handle large amounts of data.

Currently, the most common Wi-Fi standard is used in homes and businesses, providing transfers in the area of hundreds of megabits per second. This is good for file transfers but can be too demanding for many IoT applications.

Wi-Fi has a range of approximately 50 meters and is currently mostly used in the home. Wi-Fi is built on the 802.11 standards, and the latest version is 802.11/ac.

Benefits of Wi-Fi are universal smartphone compatibility that is affordable, well protected, and controlled. Disadvantages of Wi-Fi are relatively high power consumption, instability, and inconsistency.

NFC

NFC (near-field communication) is a technology that enables easy and secure two-way transfers between electronic devices, especially for mobile phones, enabling consumers to make wireless payment transactions, access digital content, and connect to electronic devices. Essentially, the capacity of wireless card technology is expanded and enables devices to share information at less than 4 cm.

Near-field communication uses electromagnetic induction between two antennas located within each other's adjacent fields, effectively forming an air-core shaped transformer. NFC handles stickers, car keys, or battery-powered cards. NFC peer-to-peer communication is possible if both devices are powered.

NFC offers a low-speed connection with a single setup. NFC has a short range and supports encryption.

A disadvantage of NFC is its short range, and it may not be possible to use it in many situations, as it is currently available only on new Android phones and at Apple Pay on new iPhones.

NFC devices can be used in wireless payment systems, such as those used in credit cards and electronic cards, and enable mobile payment by replacing or completing these systems.

Sigfox is an alternative broadband technology, which in terms of range lies between Wi-Fi and mobile. It uses the industrial, scientific, and medical (ISM) radio bandwidth that is free to use, without having to obtain licenses, to transfer data over a very narrow range to and from connected objects. The idea of Sigfox is that many M2M applications running on a small battery require only low data transfer. This is ideal if the distance is too small for Wi-Fi and the mobile is too expensive and consumes too much power. Sigfox uses a technology called ultra-narrow band (UNB) and is designed to handle only low data transfer rates of 10 to 1000 bits per second.

Already used for tens of thousands of connected objects, the Sigfox network is currently being rolled out in major cities across Europe. The network offers a robust, energy-efficient, and scalable network that can communicate with millions of battery-powered devices over several square kilometers, making it suitable for various M2M applications, which are expected to include smart meters, patient monitors, security devices, streetlights, and environmental sensors.

The frequency used is 900 MHz. The range is 30-50 km in rural areas and 3-10 km in cities. The data rates are 10-1000 bps.

Neul has a similar concept to Sigfox and operates below the 1 GHz band. Neul utilizes very small portions of the TV white space spectrum to deliver high scalability, high coverage, low power consumption, and low-cost wireless networks. The systems are based on the Iceni chip, which communicates using white space radio signals to access the high-quality UHF spectrum, now available due to analog to digital television transition.

The communication technology is called Weightless, a new wireless networking technology for broadband designed for the IoT, which is highly competitive with existing GPRS, 3G, CDMA, and LTE WAN solutions. Data rates can range from a few bits per second up to 100 kbps over the same single link, and devices can consume as little as 20 to 30 mA from two AA batteries, meaning 10 to 15 years in the field.

The frequencies are 900 MHz (ISM), 458 MHz (UK), or 470-790 MHz (white space). The range is 10 km, and the data rates are a few bps up to 100 kbps.

LoRa is a long-range wireless communication protocol. Since LoRa defines the lower physical layer, the upper networking layers are lacking. LoRaWAN was developed to define the upper layers of the network. LoRaWAN is similar to Sigfox and Neul and is designed to provide low-power WAN functions, which are particularly needed to support low-cost mobile phones with secure two-way communication in the IoT, as well as machine-to-machine communication in smart cities and industrial applications. Optimized for low power consumption and large network support with millions and millions of units, LoRa transfers data volumes ranging from 0.3 kbps to 50 kbps.

The range is 2-5 km in urban areas and 15 km in inland countryside.

Which Standard Will Be Common in the Future?

It is highly likely that the winner of these standards will be a standard that is available in many of the new devices and phones; otherwise, people will not use it. Today, every smartphone has Bluetooth and Wi-Fi. However, NFC is increasingly being implemented in new phones.

Experience shows that a standard arises as a clear winner if it has a well-defined application. For example, if you want to transfer large amounts of files, Wi-Fi is ideal. If you want to communicate at a short distance, there is nothing like Bluetooth. If you want fast, short-term interaction, NFC can be the best. Furthermore, the winning communication protocol really depends on your specific goals and your clearly defined usage case.

16 PLATFORMS FOR THE INTERNET OF THINGS

Developing for the IoT is a complex task, and nobody wants to do it from scratch. IoT data platforms offer a jumping-off point by combining many of the tools needed to manage a deployment from device management to data prediction and insights into a single service.

An IoT platform is a multilayer technology that enables straightforward provisioning, management, and automation of connected devices within an IoT system. It connects your hardware, however diverse, to the cloud by using flexible connectivity options, enterprise-grade security mechanisms, and broad data processing powers. For developers, an IoT platform provides a set of ready-to-use features that greatly speed up the development of applications for connected devices while taking care of scalability and cross-device compatibility.

IoT platforms can help in the following activities:

- To connect hardware, such as sensors and devices
- To handle different hardware and software communication protocols
- To provide security and authentication for devices and users
- To collect, visualize, and analyze data that the sensors and devices gather
- To integrate all of the above with other web services

IoT platforms are the support software that connects everything in an IoT system. An IoT platform facilitates communication, data flow, device management, and the functionality of applications.

IoT platforms originated in the form of IoT middleware, whose purpose was to function as a mediator between the hardware and application layers. Its primary tasks included data collection from the devices over different protocols and network topologies, remote device configuration and control, device management, and over-the-air firmware updates.

16.1 SOME IOT PLATFORMS

There are now several hundreds of IoT platforms in the rapidly changing platform vendor market, and it is pushing toward 1,000 platforms soon. The IoT platform market is exploding at a compound annual growth rate of 39% and is expected to surpass 22 billion dollars by 2023.

Figure 16.1 Some IoT platforms (IoT Analytics).

The following sections give a description of some IoT platforms (in alphabetical order).

Google Cloud IoT

Google Cloud IoT is a complete set of tools to connect, process, store, and analyze data both at the edge and in the cloud. The platform consists of scalable, fully managed cloud services; it is an integrated software stack for edge/on-premises computing with machine learning capabilities for all your IoT needs.

Kaa Platform

Kaa is an enterprise-grade IoT platform built on a modern cloud-native architecture and a fully customizable feature set. Based on flexible micro services, Kaa easily adapts to almost any need and application. It scales from a tiny start-up to a massive corporation and supports advanced deployment models for multi cloud IoT solutions. However, you can also use it to put together a smart thermostat for your living room.

Microsoft Azure IoT Suite

Microsoft Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers. It provides software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) and supports many different programming languages, tools, and frameworks, including both Microsoft-specific and third-party software and systems.

ThingSpeak

ThingSpeak is an IoT analytics platform service that enables you to aggregate, visualize, and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MatLab code in ThingSpeak, you can perform online analysis and processing of the data as it comes in. ThingSpeak is often used for prototyping and proof-of-concept IoT systems that require analytics.

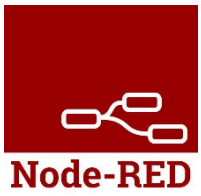
ThingWorx

The ThingWorx platform is a complete, end-to-end technology platform designed for the IIoT. It delivers tools and technologies that empower businesses to rapidly develop and deploy powerful applications and augmented reality (AR) experiences.

17 WIRING THE INTERNET OF THINGS

Wiring together things in the IoT can be challenging and laborious. For that reason, tools have been developed for helping you to do this.

One of the challenges of the IoT is stepping into an object-oriented world and understanding how to link together so many disparate objects that speak different languages. The average IoT engineer is not interested in diving into the coding necessary to drive these interactions but wants to be able to pull in operational data quickly. This is why a tool that makes it easy to wire together the IoT is incredibly valuable.



Node-RED is a visual tool for connecting things to each other in the IoT. Node-RED is a visual programming development tool developed by IBM to connect hardware devices, APIs, and online services as part of the IoT.

Node-RED is a programming tool for connecting hardware devices, APIs, and online services in new and interesting ways. It provides a browser-based editing tool that makes it easy to link things using a wide variety of nodes in the tool palette.

Node-RED has a browser-based editor that can be used to create JavaScript functions. Items of applications can be stored or shared for reuse. Data streams created in Node-RED are stored using JSON.

Node-RED reduces the need to write code, lowering the technical bar and enabling those interested in developing for the IoT to focus on the creating rather than on the doing. People use Node-RED for a surprising variety of applications, including schools teaching kids to code using Node-RED due to its ease-of-use.

The beauty of Node-RED is that almost anyone can quickly learn to use it; it is not limited to the realm of programmers. It can be used just as easily on a Raspberry Pi as in cloud environments, such as IBM Bluemix.

Moreover, while Node-RED is an incredibly useful tool for wiring together the IoT, it has applications far beyond the IoT. It can be used as a generic event-processing engine. For example, you can use it to listen to events from HTTP, web sockets, TCP, and Twitter and then capture and store that data. You can also use it to implement simple REST APIs. You can do all of this without having to program much at all.

18 INTERNET OF THINGS SECURITY

The IoT affects all aspects of our lives. We use the IoT in our homes, offices, and cars. The advantage is that this has enabled us to do things we have not been aware of before, but there is also a major disadvantage of the IoT. The IoT is becoming an increasingly attractive target for cybercriminals. Multiple connected devices mean more attack angles, and these provide more opportunities for attackers. This has caused a security issue, and unless we do something to tackle this ever-growing problem, it will have serious consequences.

Security is a problem with the IoT. In many cases, IoT sensors collect sensitive data, such as what people say and do in their own homes. This means that safety is important when using the IoT, but until now IoT safety has been very poor. Many IoT embedded systems have been designed with little thought for basic security.

The IoT bridges the gap between the digital and the physical world, which means that hacking of devices can have serious consequences in the real world. If hackers manage to take control of a car without a driver, it can end in an accident. Hacking sensors that control the temperature of a nuclear power plant may cause the operators to make a wrong and catastrophic decision.

Software errors have made smart home appliances such as refrigerators, ovens, dishwashers, and webcams accessible to hackers, which can use these items as an entry point to a network or to send spam. As the cost of creating smart IoT things decreases, these issues will only become more and more widespread and difficult to handle.

Errors are constantly being discovered in software, but many IoT devices lack the ability to be upgraded, which means they will be constantly in danger. Attacks from hackers are now targeting IoT devices such as routers and webcams because these devices are deficient in security, making them easy to attack.

It has been demonstrated that cars connected to the Internet can be attacked and that hackers can perform any type of unfortunate activity, including taking control of the entertainment system or unlocking or closing the doors of a driving car. Another type of IoT hacking that is a great cause for concern is the hacking of medical devices, which can have harmful and possibly fatal consequences for patient health.

There are increasing numbers of connected IoT devices, and that means more attack angles and more opportunities for hackers to attack. Security experts have warned of the potential risk of the many unsecured devices that have been connected to the Internet.

The idea of linking IoT devices and other objects to the Internet is relatively new. Therefore, safety has not always been taken into account when designing IoT products. IoT products are often sold with old and unpacked embedded operating systems and software. Furthermore, buyers often fail to change the default passwords of smart devices, or if they change them, they do not use sufficiently strong passwords. To improve security, an IoT device that needs to be directly accessible over the Internet must be segmented into its own network and given limited access to the network. The network segment must then be monitored to identify potential irregular traffic, and something must be done immediately if there is a problem.

IoT weaknesses have created new opportunities for hackers. The vulnerabilities found on IoT devices are serious and have made IoT security a problem that needs to be resolved quickly.

18.1 SOME THREATS TO IOT SECURITY

In the news about new technology, you can read about the hacking of businesses, the stealing of identities, and the hijacking of app-connected cars.

These events give reason to worry. It is easy to understand how digitally related things have a certain security risk. Often, the default settings of the device are similar to wide-open settings, and many organizations do not have proper security protocols.

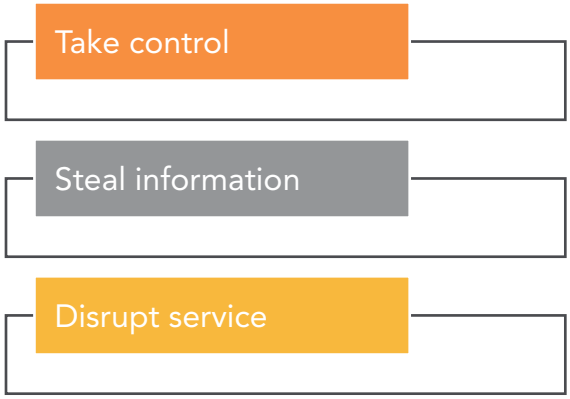


Figure 18.1 Security risks have significantly increased with the rise of connected IoT devices. Three types of IoT attacks are to take control, to steal information, or to disrupt service.

To take control: Controls in homes for smart door locks and lighting systems can be vulnerable. Door locks in cars can be opened remotely.

To steal information: Personal fitness devices can tell a hacker where you are. IoT devices in your home can give information about your personal life.

To disrupt service: Hacked vehicle control systems can allow the remote control of brakes. Peacemakers can be attacked remotely.

There are five main types of IoT attackers today:

1. Amateur hackers, such as script kids and hobbyists
2. Criminals, such as low-level cybercriminals
3. Cyber espionage groups, such as organized syndicates or criminal groups
4. Terrorists or political hack activists
5. Foreign espionage, such as state-financed attackers

Each type of attacker may have different talents and aims, either as an individual or as a group. Given the same tools, different classes of attackers can achieve different outcomes.

However, cyber-espionage groups with large resources and highly qualified petty criminals are the most common type of IoT attackers. In many cases, they have developed advanced malware with the ability to mutate and avoid detection in IoT networks, or they exploit DDoS attacks as a means of extortion.

Armada Collective is an example of a traditional cyberespionage group that has demanded businesses to pay thousands of dollars to them or run the risk of their services being disrupted due to cyberattacks. Although the original members of the Armada Collective appear to be in prison now, some people who are motivated by financial gain continue to use the group's name for extortion.

18.2 SECURITY CHALLENGES

Any distributed system is expected to be well secured and reliable and to meet the clients' privacy criteria. For the defense and the medical sector, a security system is the first thing to consider. IoT systems have many sensitive applications in governments and personal lives. However, we never can say that the environment is completely secure.

The following key challenges in IoT security can be identified:

1. Most of the IoT devices are extremely small. It is very hard to add an extra security module to those tiny things.
2. Most of the IoT things have very low computational capabilities. So many complex security algorithms are not suitable for them.
3. Limited power in IoT things is the most challenging barrier to IoT security. Any extra security module, whether software or hardware, needs extra energy to perform. However, IoT systems, especially the wireless systems, are always expected to be energy efficient.
4. The software of things, in most cases, cannot be updated. So a presently secure device may become insecure after a few years.
5. The industry has not taken physical layer security issues seriously until now. However, hardware-level threats are increasing exponentially in electronic devices.

Every IoT device has privacy and security issues. These issues range from hackers who steal our data and even threaten our lives to businesses that can easily acquire private data. Since the progress of the IoT will not stop, these are major issues that consumers and businesses must consider before using devices connected to the Internet.

Security and privatization are critical issues facing the development of the IoT. Below, we will look at some of the challenges that are central to making the IoT safer.

More Devices Mean More Hacking Opportunities

The basic security issue of the IoT is that the number of devices increases behind the network firewall. Ten years ago, we had to worry about protecting our computers. Five years ago, we also had to worry about protecting our mobile phones. Now we have to worry about protecting our car, our household appliances, our wearables, and a host of other IoT devices.

Because there are so many devices that can be hacked, hackers can achieve more. You may have heard of how hackers could remotely control cars. However, hackers can even use seemingly insignificant IoT devices such as baby monitors or thermostats to reveal private information or just ruin your day. The point is that we need to think about what a hacker can do with any entity if he or she can break through the security.

Need for Updates

Although you may be good at properly configuring a connected IoT device, there are other loopholes. Manufacturers of connected devices are often slow in releasing updates. Some companies do not provide support at all. Instead of updating previous devices, they prefer to fix security issues with the next version they make of the thing. Security and privacy on the IoT are therefore the responsibility of the user.

As the IoT is growing, we must worry about protecting more and more devices. However, even if you start to take security seriously, it is the technology companies that should make these new units secure. In addition, the problem is that these companies do not update their devices well enough or not at all. This means that an IoT device that was safe when you bought it could become insecure after hackers discovered new security holes.

Computers used to have this problem, but automatic and easier updates have helped alleviate the problem. Computers have automatic updates, in part because most users are too lazy to perform even the basic steps needed to keep their computers safe. Moreover, when you consider that the protection of the myriad IoT devices will be much more difficult than a single computer, this problem will be even worse.

Manufacturers Can Be a Security Threat

Hackers are scary, but they are far from the only threat to the IoT. Backdoors are discovered in software. In fact, companies that make and distribute interconnected devices can also use these devices to acquire personal data, which is particularly dangerous for money transfers.

The pressure with which manufacturers must get their IoT units on the market often results in compromising on security. Although they can offer firmware upgrades for some time, they often stop doing this as they start to focus on building the next device, thus giving customers a bit of outdated hardware that could become a security risk.

Privacy and the Internet of Things

The fact that sensors collect data on everything you do means that the IoT is a potential headache for privacy.

For example, smart homes can tell when you wake up, when your smart coffee machine is activated, how well you brush your teeth thanks to your smart toothbrush, which radio station you are listening to thanks to your smart speaker, the type of food you eat thanks to your smart oven or fridge, what the kids are thinking thanks to their smart toys, and who visits you and walks past your house thanks to your smart door cameras.

The safety of IoT data is important for privacy. It is surprisingly easy to find out much about a person from a few different sensor readings. In one project, researchers found that by analyzing data mapping all day just home energy consumption, carbon monoxide and carbon dioxide, and temperature and humidity, they could figure out what someone had for dinner.

Consumers need to understand the safety risks of using IoT devices, and whether they are happy with it. Some of the same issues apply to businesses. Will your management team like to discuss a merger in a meeting room equipped with smart speakers and cameras, for example? A recent study found that four out of five companies could not identify all their IoT units on the network.

Physical and Hardware Security

The security issues in the IoT are not restricted to data authentication, access control, client privacy, and other attacks such as data leakage. Hardware-level insecurity is also grabbing the attention of researchers nowadays, and it is becoming a growing problem day by day. To get a completely hardware-secured IoT system, we need to secure the integrated circuits in the IoT-enabled devices.

Many IoT deployments will be in remote and isolated areas, leaving sensors and edge routers vulnerable to physical attack. It does not help that data security is very good if someone can get access to or steal sensors and equipment.

18.3 MORE LAYERS OF IOT SECURITY ARE NEEDED

The IoT poses many security threats. A big difference between the IoT and previous Internet technologies is that the number of possible threats is much greater due to the following:

1. Having more and more IoT devices means increasing the number of attacks.
2. IoT technology has many attack points. Hackers can attack IoT devices, connectors, network communications, and data centers.
3. IoT technology is becoming increasingly complex, giving new opportunities for attack.
4. Attacks on IoT devices can have very serious consequences. For example, hacking a car can lead to an accident.

Architectures for the IoT require multiple layers of security that work together to provide complete end-to-end security from device to cloud and for the entire life cycle of an IoT product.

IoT security occurs on four different layers, as outlined in Fig. 18.2.

Need for security	
The device layer	The device layer refers to the hardware of the IoT solution, that is, the physical thing. Security includes physical security, stored data, chip security, secure start-up, device authentication, and each device having an identity.
The communication layer	The communication layer refers to the networks that are connected to the IoT solution, that is, the media over which the data is transmitted. Security components include access control, firewall, IPS, IDS, and end-to-end encryption.
The cloud layer	The cloud layer refers to back-end software for the IoT solution, that is, the data center to which data from IoT devices is sent and in which data is analyzed and interpreted to gain insights to be able to perform actions. IoT cloud providers are expected to provide secure and efficient cloud services that protect data and prevent downtime.
Life-cycle management	Secure life-cycle management refers to a parent layer of continuous processes required to keep the security of an IoT solution up to date. Safety components include risk assessment, activity monitoring, updates, vendor control, user awareness assessment, and safe settlement.

Figure 18.2 The table shows the four layers of IoT security.

With the expected growth of millions of IoT devices, it will not be possible to handle security updates and security issues for IoT devices manually. There is therefore a need for automation of security tasks. Security automation techniques that merge security solutions and artificial intelligence are becoming increasingly widespread.

There is no single IoT security provider that can offer a complete, end-to-end protected security solution. However, some companies offer more than others offer and can provide a pretty good and complete end-to-end IoT security solution.

18.4 HOW TO SECURE THE INTERNET OF THINGS?

Doing something about IoT device security issues is not enough. For example, you must also secure gateways that connect IoT devices to networks.

IoT devices are always connected and active. Unlike devices that require a login to use, IoT devices go through only one authentication process once. This can make them the perfect target for penetrating a corporate network. Therefore, you must ensure the security of the gateway to improve the overall security of IoT systems.

Below, we will discuss some technologies for IoT security.

IoT Network Security

There is a need to protect and secure the network that connects IoT devices with back-end systems on the Internet.

Securing an IoT network is much more challenging than securing traditional networks. The reason is more complexity because there are many communication protocols, standards, and entities involved. Hackers would like to prioritize attacking networks because it will give them control over all the IoT units in the network. Therefore, to prevent and detect intrusions, you should use antivirus, firewalls, and other systems to secure an IoT network.

Using strong encryption to secure protocols is another good network security practice. Communication between devices can potentially be hacked, and both the IoT and the IIoT use several multilayered network protocols. The use of encryption in both network layers and transport layers can provide a good barrier to online attacks.

IoT Authentication

One way to secure an IoT device from attacks is by authentication. In the field of computer science, authentication means verifying the digital identity of a sender attempting to communicate.

Users can use simple authentication or more complex authentication. More secure forms of authentication are two-way authentication, digital certificates, and biometrics.

Unlike traditional authentication methods that require the presence of a human being, IoT authentication does not require human actions, as the IoT involves mainly built-in sensors and machine-to-machine interactions. Therefore, a completely different way of thinking must be used when authenticating IoT devices.

IoT Encryption

Many connected devices store and transmit sensitive, personally identifiable information, and this data must have good protection. Both requests sent over the web and user data located on a file must be encrypted.

Encryption of stored data and data in motion will help maintain the integrity of the data and reduce the risk of the data being sniffed by hackers. There are different hardware profiles on different IoT devices, so no standard protocol and encryption can be implemented across all IoT devices. This presents a great challenge when it comes to encrypting IoT data, as different encryption techniques must be used for different devices.

The positive thing about encryption is that if you can encrypt your IoT data, you can avoid data being read or modified.

IoT PKI

Public key infrastructure (PKI) is a framework for issuing, administering, and using digital certificates over computer networks. Application areas for PKI are encryption, authentication, and signing of documents or software.

PKI provides complete X.509 digital certification, encryption key, and life-cycle features, including public/private key generation, deployment, management, and recall. However, hardware specifications for some IoT devices may limit or prevent the ability to utilize PKI.

Digital certificates can be securely loaded onto IoT devices when they are created and can then be enabled by third-party PKI software packages. The certificates can also be installed after IoT devices have been manufactured.

IoT API Security

An application programming interface (API) is a set of clearly defined communication methods between different software components. A good API makes it easier to develop a computer program by offering all the building blocks.

REST (representational state transfer) is an architectural style that has proven to be well suited for developing distributed media applications. Although REST is generally applicable, it is most commonly used in connection with communication with services via HTTP.

Documented REST-based APIs provide the ability to authenticate and authorize data streams between IoT devices, back-end systems, and applications. API security will be critical to protecting the integrity of data transfers between edge units and back-end systems to ensure that only authorized devices, developers, and applications communicate with APIs and to detect potential threats and attacks against specific APIs.

A Combination of Different Security Technologies Is Best

It is important to know that IoT devices can be an easy target for hackers and cybercriminals who want to enter the IoT network and access confidential and sensitive information. You must be aware that there is no easy way to fix all IoT security issues.

To protect IoT networks from external attacks, it is beneficial to use a combination of different technologies. Authentication and encryption are central, and there are measures to secure the network. Establishing an efficient public key infrastructure will help manage the entire process from key generation to key storage efficiently.

19 DESIGN FOR THE INTERNET OF THINGS

The IoT has a large market. Everything from cars to refrigerators can be connected to the Internet.

However, few companies have the expertise or resources to build all parts of an IoT design from scratch. Online providers offer a variety of solutions to help designers add wireless connectivity to new or existing products. Developers, nevertheless, often hesitate to use new technologies, as they fear that the company lacks experience in the area and are uncertain about which technologies they should focus on.

A procedure called virtual prototyping is a good way to start an IoT project. Virtual prototyping enables designers to make the right technology choices and to define and test system attributes.

A prototype is a preliminary edition of a product. The purpose of a prototype is to demonstrate and test function and design. In this way, cost limitations, time to marketing, and performance requirements can be evaluated and balanced to ensure that the end product is both valuable and affordable before making major investments in design or production.

19.1 UNDERSTAND THE APPLICATION

While engineers can connect different components to an IoT product, expertise is required to take all the necessary steps to get an IoT product on the market. Before starting to design a project, it is important to understand fully the hardware, software, and certification requirements of the project.

There are several important questions to ask in this analysis phase.

What Is the Aim of the Project?

What should the project be used for? Does it need to work in real time? Should it be used for automation or control? A good understanding of how a project works and what the project will be used for allows you to determine the required power and performance levels.

What Are the Requirements for Size?

Often, it is desired that IoT units be as small as possible, but this can quickly become expensive. Take, for example, Wearables. The performance of these portable IoT devices is often limited, and they are capable of supporting only small amounts of data. If small size and high performance is the goal, a long-life battery will be needed to meet power consumption, which can quickly make the overall solution bigger and more expensive.

What Are the Requirements for the Communication Distance?

Inside a house or in a city with easily accessible Wi-Fi, the range can be measured in feet or meters. However, outdoors or in a rural area, the range required may be several kilometers for a signal to reach the nearest server or wireless gateway, requiring a mobile or GPS interface.

If there is a long transmission distance, higher power and higher frequencies will be needed. If the location is remote and cannot be reached easily, the battery life will be important. Disturbances with physical obstructions or other RF devices can also affect the operating distance.

What Is the Power Source?

The power source is a critical component in designing an IoT application, since it affects both the distance of communication and the battery life. The longer the range, the more power is needed. The more power is required, the shorter the battery life. If the device is to be powered by batteries alone, it is only necessary to design with the intention of saving power.

Typically, a lithium-ion battery is the standard form of power in mobile devices due to its energy density.

Batteries lose capacity with many charge-discharge cycles. This capacity loss is expressed as a measure of the initial capacity, for example, 30% loss after 1000 cycles. The rate of loss depends on chemistry and temperature. Typically, a lithium-ion battery can last for 10 years, whereas an alkaline battery will last for only 5 years.

Many network technologies will not work well with battery power. The frequency of communication will also influence the choice of power supply.

Environmental Considerations

One of the many benefits of wireless systems is that they can often work where people cannot, including inaccessible or dangerous environments. However, it is important to control the types of wireless systems that can operate in specific environments. You must take into consideration whether the environment is hot, cold, wet, or dry.

For example, an IoT device to be used in a freezer or a device used in a high-heat location will likely require more frequent monitoring and possibly a built-in emergency alert.

Communication

Should the device we make be able to communicate with other devices? In that case, one must ensure that they can cooperate. This goes beyond adhering to standards from organizations such as IEEE, ISO, and others because even these known standards can be interpreted in more than one way.

Safety

Even if your project is not aimed at uses in health organizations, the financial industry, or the military, information security will be a major consideration in the design of your project. One should design as many layers of encryption protocols as possible, at least transport layer security (TLS) and passwords.

20 STATISTICS ABOUT THE INTERNET OF THINGS

20.1 THE SIZE OF THE IOT

The number of things connected to the Internet continues to increase. There are already more things connected to the Internet than there are people in the world. Every second, another 127 devices are connected to the Internet.

Everyday objects are increasingly connected to the Internet, ranging from smart refrigerators to smart toothbrushes. In the coming years, the number of smart units in our homes will only increase further. The consulting company Gartner anticipates that a regular home could contain more than 500 smart units by 2022.

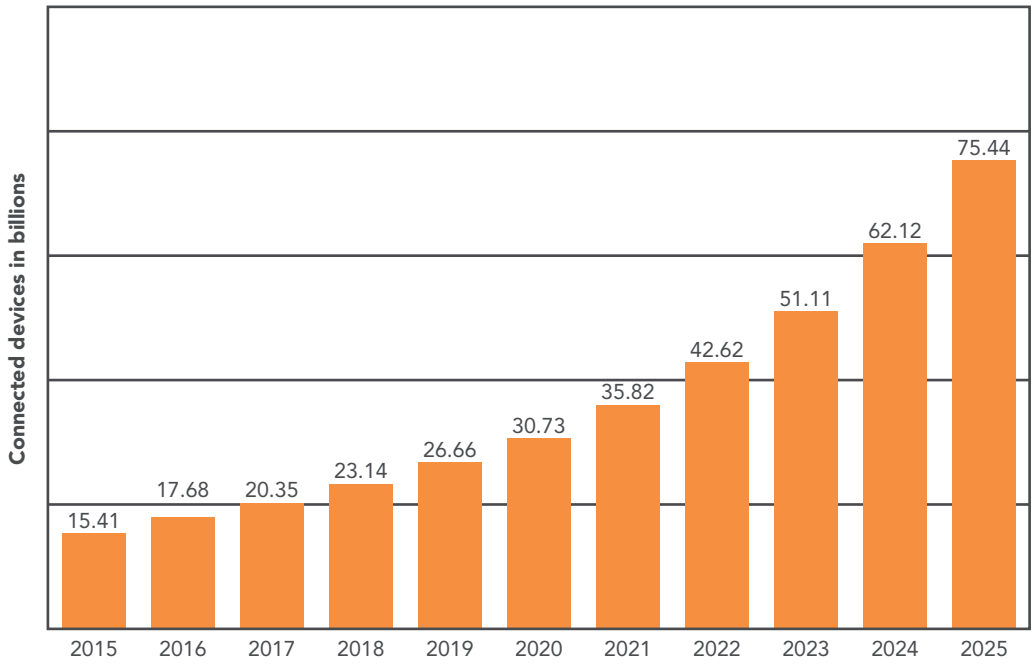


Figure 20.1 The figure shows connected IoT devices worldwide in billions from 2015 to 2025 (Statista).

IoT devices are becoming increasingly cheaper as the prices of sensors and communication via the Internet continues to fall. The sensors are getting smaller, better, and cheaper every year. The cloud that provides the necessary infrastructure is easily accessible and relatively affordable. This provides a good basis for the IoT to grow in the years to come.

The possible uses of the IoT are virtually endless. Anything or any device can become smart if it connects sensors to the device and if it is connected to the Internet. When machine learning or other software designed to analyze data is used, the data from the sensors can provide valuable insights for people.

With so many opportunities, cheap infrastructure, and high demand, we can expect an explosion of connected IoT units in the coming years. It looks like we are moving toward a world that is fully automated.

20.2 HOW MUCH MONEY IS INVOLVED IN THE IOT?

In 2016, global spending on the IoT across markets was 737 billion U.S. dollars. The IDC (International Data Corporation) predicts that by 2020, this number will reach 1.29 trillion U.S. dollars, a compound annual growth rate of 15.6%.

According to General Electric, the IoT will add 10 to 15 trillion U.S. dollars to worldwide GDP (gross domestic product) growth by 2030, which is the equivalent of China's entire current economy.

The healthcare market will feel the impact of the IoT. ACT, an organization representing over 5,000 mobile technology companies, is forecasting a U.S. 117 billion dollars connected healthcare market by 2020.

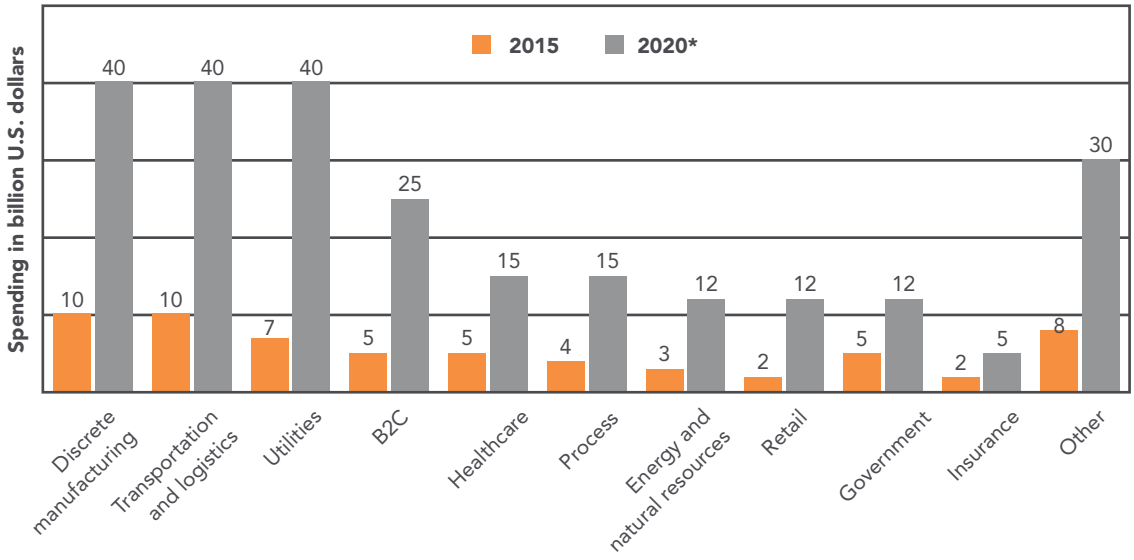


Figure 20.2 Spending on the IoT worldwide in 2015 and 2020 in billion U.S. dollars (Statista).

The IoT can also save money. For example, the city of Barcelona saves 37 million U.S. dollars a year, thanks to smart lighting. In addition, the city's many IoT initiatives have created 47,000 new jobs.

20.3 WHAT IS THE FUTURE OF THE IOT?

This section is based on Gartner's predictions for the future of the IoT. Gartner is the world's leading research and advisory company.

More Sensors

Embedded sensors are used to collect data from around the world. The IoT sensor market alone is expected to be worth a staggering 27 billion U.S. dollars by 2022. That means that we will see many more sensors in the future.

The sensor market will evolve continuously until at least 2023. New sensors will enable a wider range of situations and events to be detected, current sensors will fall in price to become more affordable or will be packaged in new ways to support new applications, and new algorithms will emerge to deduce more information from current sensor technologies.

Artificial Intelligence and Machine Learning

Artificial intelligence (AI) means that machines can perform tasks in ways that are intelligent. Machine learning is technically a branch of AI. Machine learning is based on the idea that we can build machines that can process data and learn on their own, without human supervision.

Machine learning is going to become an integral part of IoT devices. Big companies such as Microsoft, IBM, and Google are investing in AI and machine learning. It is the only way to move forward.

The technology landscape for AI is complex and will remain so until 2023, with many IT vendors investing heavily in AI, variants of AI coexisting, and new AI-based tolls and services emerging. Despite this complexity, it will be possible to achieve good results with AI in a wide range of IoT situations.

AI will be applied to a wide range of IoT information, including videos, still images, speech, network traffic activity, and sensor data.

High-profile examples of AI include autonomous vehicles (such as drones and self-driving cars), medical diagnosis, creating art (such as poetry), proving mathematical theorems, playing games (such as chess or Go), search engines (such as Google Search), online assistants (such as Siri), image recognition in photographs, spam filtering, predicting flight delays, prediction of judicial decisions, and targeting online advertisements.

The Shift from Intelligent Edge to Intelligent Mesh

The shift from centralized and cloud architectures to edge architectures is well underway in the IoT space. However, this is not the end because the neat set of layers associated with edge architectures will evolve to a more unstructured architecture comprising a wide range of things and services connected in a dynamic mesh.

These mesh architectures will enable more flexible, intelligent, and responsive IoT systems, although often at the cost of additional complexities.

Trusted Hardware and Operating Systems

Gartner surveys invariably show that security is the most significant area of technical concern for organizations deploying IoT systems. This is because organizations often do not have control over the source and nature of the software and hardware being utilized in IoT initiatives.

However, by 2023, Gartner expects to see the deployment of hardware and software combinations that together create more trustworthy and secure IoT systems.

Silicon Chip Innovation

Currently, most IoT endpoint devices use conventional processor chips, with low-power ARM architectures being particularly popular. However, traditional instruction sets and memory architectures are not well suited for all the tasks that endpoints need to perform. For example, the performance of deep neural networks (DNNs) is often limited by memory bandwidth rather than processing power.

By 2023, it is expected that new special-purpose chips will reduce the power consumption required to run a DNN, enabling new edge architectures and embedded DNN functions in low-power IoT endpoints. This will support new capabilities such as data analytics integrated with sensors, and speech recognition included in low-cost battery-powered devices. This trend of incorporating silicon chip functions such as embedded AI will enable organizations to create highly innovative products and services.

New Wireless Networking Technologies for the IoT

IoT networking involves balancing a set of competing requirements, such as endpoint cost, power consumption, bandwidth, latency, connection density, operating cost, quality of service, and range. No single networking technology optimizes all of these, and new IoT networking technologies will provide additional choice and flexibility. In particular, they should explore 5G, the forthcoming generation of low-Earth-orbit satellites and backscatter networks.

Tightened Security and Higher Physical Standards

Hacks and data breaches have proved that the security of IoT devices needs to be much better. We will also see changes in physical standards. In the future, devices will be valued based on their level of security and endurance rather than on how cheap they are to produce.

Data Privacy Will Become a Priority

When it comes to the IoT, the main concern of consumers and developers is privacy. IoT products are going to be strictly regulated, and privacy will be the primary concern in the future.

REFERENCES

Bose, A. Embedded system – Characteristics, types, advantages & disadvantages. Retrieved January 2019 from https://electricalfundablog.com/embedded-system-characteristics-types-advantages-disadvantages/

McClelland, C. What is an IoT platform? IoT for all. Retrieved January 2019 from https://www.iotforall.com/what-is-an-iot-platform/

O'Connor, C. Wiring the Internet of Things. IBM (2016). Retrieved January 2019 from https://www.ibm.com/blogs/internet-of-things/wiring-internet-things/

Embedded system. (2019). Retrieved 2019 from https://en.wikipedia.org/wiki/Embedded_system

Embedded systems design. (2012). Retrieved January 2019 from https://www.sciencedirect.com/topics/computer-science/embedded-system-design

Embedded systems tutorial. (2019). Retrieved January 2019 from https://www.tutorialspoint.com/embedded_systems/index.htm

Gartner identifies the top 10 strategic IoT technologies and trends. Retrieved May 2019 from https://www.gartner.com/en/newsroom/press-releases/2018-11-07-gartner-identifies-top-10-strategic-iot-technologies-and-trends

Singh, H. Statistics that prove IoT will become massive from 2018. (2018). Retrieved March 2019 from http://customerthink.com/statistics-that-prove-iot-will-become-massive-from-2018/

Internet of Things. (2019). Retrieved April 2019 from https://en.wikipedia.org/wiki/Internet_of_things

Internet of Things (IoT) tutorial. (2019). Retrieved March 2019 from https://www.tutorialspoint.com/internet_of_things/

Internet of Things - Statistics & facts. (2019). Retrieved March 2019 from https://www.statista.com/topics/2637/internet-of-things/

Introduction to transducers, sensors, and actuators. (2011). Retrieved January 2019 from http://www.ieec.uned.es/investigacion/Dipseil/PAC/archivos/More%20on%20Transducers%20Sensors%20and%20Actuators.pdf

Lee, J. How mesh networking will make IoT real. (2018). Retrieved May 2019 from https://hackernoon.com/how-mesh-networking-will-make-iot-real-b5b88baab63b

Fuller, J. K. The 4 stages of an IoT architecture. Retrieved March 2019 from https://techbeacon.com/enterprise-it/4-stages-iot-architecture

Gyarmathy, K. 5 IoT statistics you need to know in 2019. Retrieved May 2019 from https://www.vxchnge.com/blog/iot-statistics

Prokopets, M. Ultimate list of 30 IoT platforms you must try in 2018. (2018). Retrieved June 2019 from https://dzone.com/articles/ultimate-list-of-30-iot-platforms-for-your-iot-pro

Hasan, M. Top 15 best embedded systems programming languages. (2012). Retrieved January 2019 from https://www.ubuntupit.com/top-15-best-embedded-systems-programming-languages/

Hegdes, M. Cloud storage vs local storage – Which is right for your Business? Retrieved May 2019 from https://www.contegix.com/blog/cloud-storage-vs-local-storage-which-right-your-business

Sethi, P., and Sarangi, S. R. Internet of Things: Architectures, protocols, and applications. Retrieved April 2019 from https://www.researchgate.net/publication/312957467_Internet_of_Things_Architectures_Protocols_and_Applications

Lea, P. (2018). Internet of Things for architects. Pact Publishing.

Koley, S., and Ghosal, P. Addressing hardware security challenges in the Internet of Things: Recent trends and possible solutions. ResearchGate (2015). Retrieved May 2019 from https://www.researchgate.net/publication/281150039_Addressing_Hardware_Security_Challenges_in_Internet_of_Things_Recent_Trends_and_Possible_Solutions

Noergaard, T. (2013). Embedded systems architecture. A comprehensive guide for engineers and programmers. Newnes

Jindal, T. Role of embedded systems in robotics. Retrieved January 2019 from http://www.robogalaxy.com/post/Role-of-Embedded-System-in-Robotics

Agarwal, T. Embedded systems role in automobiles with applications. Retrieved January 2019 from https://www.edgefx.in/importance-of-embedded-systems-in-automobiles-with-applications/

What is an IoT platform? Retrieved April 2019 from https://www.kaaproject.org/what-is-iot-platform