PART I

INTRODUCTION TO THE INTERNET OF THINGS

“*When we talk about an Internet of things, it's not just putting RFID tags on some dumb thing so we smart people know where that dumb thing is. It's about embedding intelligence so things become smarter and do more than they were proposed to do.*”

Nicholas Negroponte

1 THE INTERNET OF THINGS

The Internet has initially started as the "Internet of Computers", a global network enabling services that now include the World Wide Web (WWW), File Transfer Protocol (FTP) and others allowing computers and hence users to communicate with each other and exchange information. In the recent years, the device processing power and storage capacity are increasing while at the same time technology is making devices pervasive, mobile and wearable. In addition, networking technologies are evolving and communication electronic systems are becoming smaller and cheaper. Devices are increasingly fitted with sensors and actuators, creating environments where the former are connected to various networks. Devices can sense, compute, act and thus intelligently become parts of the so-called 'Internet of Things'.

There are several definitions for the Internet of Things (IoT) that also explain what are the main functionalities of it and what we should expect from when connecting 'Things' with each other and with the Internet. Some people suggest, that the "Internet of Things can be seen as a potentially integrated part of the 'Future Internet'. Wikipedia defines IoT as:

"A part of a dynamic global network infrastructure with self-configuring capabilities based on open and interoperable communication protocols where physical and virtual 'Things' interact with each other. These 'Things' have specific identities, physical attributes, virtual 'personalities' and use intelligent interfaces. They are able to interact and communicate among themselves and with the environment by exchanging data and information 'sensed' about the environment, while reacting autonomously to the 'real/physical world' events and influencing them by running processes that trigger actions and create services with or without our direct intervention. Interfaces in the form of services facilitate interactions with these 'smart things' over the Internet, query and change their state and any information associated with them."

To people like hobbyists, electronic enthusiasts or sensor researchers the IoT is new opportunity and at the same time a new challenge for managing the data we acquire from our embedded electronics projects and controlling their outputs.

Imaging having a small device at the size of a matchbox, that can senses temperature, humidity and light conditions of your room, and can report them directly to a web-based service. The readings by the sensors can be accessed only by you through your favorite browser, by your mobile phone and by other devices in you place, like the central heating /air conditioning system or the indoor lights control system. The latter can adjust the heat and lighting inside your place automatically, making sure you have always the most preferable conditions as defined in the web-based service by you.

Now imagine that you don't have to build everything from scratch in order to develop and deploy such a system. Imagine also that you do not have to worry about data management; how is the data stored on the web, what kind of web server and web application technology you have to use for your service, how to secure your data and implement various authentication and encryption mechanisms! You do not even have to worry about learning how to develop mobile applications that talk to your service! The web-based services that acquire and manage your sensor data, the mobile application, the communication interfaces and information exchange protocols are already available for you to exploit and explore them! What are they? They are applications and features of existing platforms that provide users with Internet of Things functionality. Where are they? We will explore a few of them and ways to use them in the following chapters of this book. Firstly, let's take a closer look at the concepts of IoT.

The Basic Concepts

Let us briefly discuss the main concepts and essential components that are mostly used in order to describe the world of the Internet of Things.

When we are referring to 'Things', we talk about devices and everyday objects, from small ones (like wrist watches and medical sensors) to really big ones (like robots, cars and buildings). All such contain devices that interact with users by generating and retrieving information about and from their environment (see Figure 1-1). They also contain hardware that allows them to control outputs (like relay switches or digital ports).

No matter what definition of the Internet of Things you may find, the main concept behind every IoT technology and implementation is the same: devices are integrated with the virtual world of the Internet and interact with it by tracking, sensing, and monitoring objects and their environment. Users and developers add components, give them sensing and networking capabilities, program them to perform the aforementioned tasks and build Web applications that interacting with the devices.

The features of a device that can act as a member of an IoT network can be summarized into the following:

- Collect and transmit data: The device can sense the environment (e.g., your home or your body) and collect information related to it (e.g., temperature and lighting conditions) and transmit it to a different device (can be your mobile phone or your laptop) or to the Internet.
- Actuate devices based on triggers: It can be programmed to actuate other devices (e.g., turn on the lights or turn off the heating) based on conditions set by you. For instance, you can program the device to turn on the lights when it gets dark in your room.
- Receive information: One unique characteristic for IoT devices is that they can also receive information from the network they belong to (i.e. other devices) or through the Internet (e.g., information from you like new triggers, new status of operation and in some cases new functionality).
- Communication assistance: IoT devices that are members of a device network can also assist in communication (i.e. data forwarding) between other nodes of the same network. Think of them as messengers for devices (nodes) that are not very close to an endpoint (e.g., your router) in order to get direct information from.

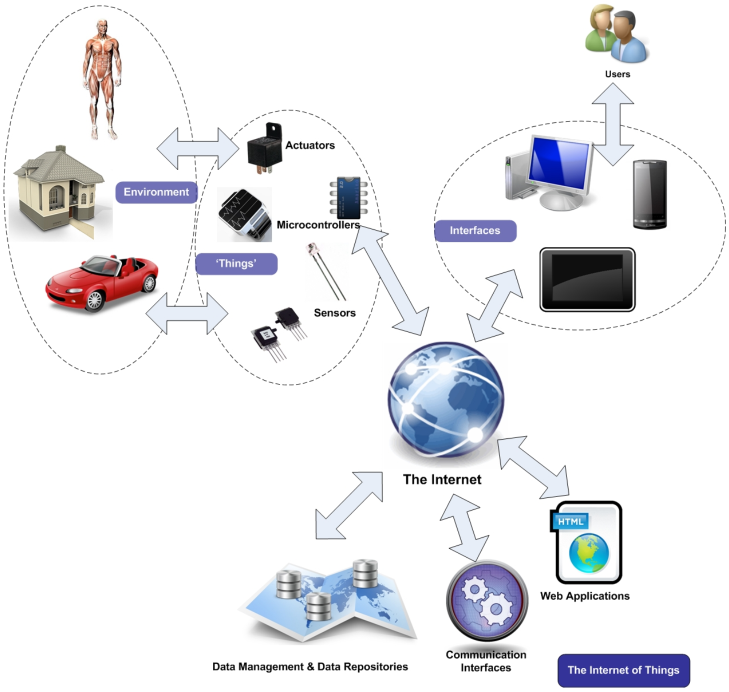

Figure 1-1. An illustration of the 'Internet of Things'. 'Things' consisting of various sensors and actuators interact with environment and the Internet allowing users to manage them and their data over various interfaces.

Interaction with the Internet

What makes IoT devices different than ordinary sensor devices is basically the ability to communicate (most usually) directly or indirectly to the Internet. So what are the main reasons a device would need to communicate with an Internet service? What kind of service would that be and what kind of features would it have? Firstly, sensors generate a lot of data that needs somehow to be managed. Usually, embedded memory is quite limited so people utilize alternative solutions like storing data on memory cards, or in computers in cases sensors are directly connected to them. Since sensors can be integrated to devices with further networking capabilities, why not to store the information online? Through this way we can solve the limited storage issue and at the same time we can access the data anywhere, anytime using appropriate web applications. Figure 1-2 illustrates the main features of 'Things' and their interaction with Internet services.

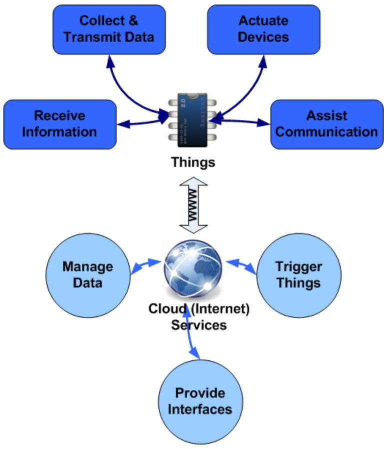

Figure 1-2. Illustration of the main features of the 'Things' in IoT networks and the respective Internet services.

In addition, most of these web applications provide interfaces for data exchange between other applications and most usefully, with mobile applications. So it is possible that your iPhone or Android mobile can talk directly to your IoT device!

We can also use the web platforms to send data back to the devices. The data can be triggers, instructions for activating actuators and switches, or simply information from the Internet, like a weather channel or a Twitter account that can displayed through an interface like an LCD screen.

The web-based platforms that enable the aforementioned functionality for data management and information exchange are based on Cloud computing. More information on Cloud computing platforms will be presented in Chapter 3 of this book.

In order for things to talk to each other and to the Internet, they need to have specific abilities and serve more specific functions. The following section discuses what are the components that are included in IoT devices that enable them to sense the environment, act and communicate.

Major Components of IoT Devices

Let's discuss what are the main components of an IoT device. Namely, such components are the main control units (the 'brains' of the devices), the sensors that collect information – signals from the environment, the communication modules and the power sources.

Control Units

Usually in the world of IoT, devices utilize a microcontroller as the main control unit responsible for the aforementioned operations. A microcontroller can be considered as a small computer on a single integrated circuit (often abbreviated as IC) containing a processor core, memory, and programmable input and output peripherals. Figure 1-3 presents a very popular microcontroller chip. The peripherals are controlled through several general-purpose input/output pins, known as GPIO pins. GPIO pins are configured to either input (read) data or output data. When they are configured to an input state, they are used to read information from sensors and accept external signals (e.g., button events). Configured to the output state, GPIO pins can control external devices such as LEDs, motors, relay switches, etc. In addition, these pins are also used by the controller to communicate with devices, such as modems and other communication modules.

Other important parts of the microcontrollers are the analog to digital converters (ADC) that are used to convert incoming analog (signal) data into a form that the processor can recognize. In addition to the converters, many such control units include a variety of timers as well.

We will discuss more on microcontrollers and their components in the following chapters.

Additional modules like Pulse Width Modulation (PWM) modules are also included in the microcontroller and enable it to process and control more advanced devices like power converters and, motors without using lots of resources in generating pulse signals programmatically.

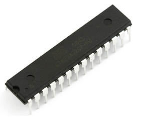

Figure 1-3. The Atmega328 chip from Atmel. One of the most widely used microcontrollers in embedded projects. Each pin has a specific function; receive analog signals, communicate through serial interface with other components, receive input voltage, generate pulses or digital output, etc. (image courtesy of Sparkfun).

Sensors

Sensors are devices that can measure a physical quantity (like temperature, humidity, etc.) and convert it into a signal, which can be read and interpreted by the microcontroller unit. They are the devices that are most likely attached to the input pins of the microcontroller in our embedded projects. Generally, most sensors fall into two categories; analog and digital sensors.

An analog sensor, such as a thermistor (which actually is a resistor changing its resistance based on temperature used in digital thermometers), or a humidity sensor, is connected into circuits so that it generates a specific voltage range, usually between 0 volts to 5 volts. This output goes to the analog input pin of the microcontroller unit (see Figure 1-4). The latter uses the appropriate analog-to-digital (A/D) circuit to convert voltage into a numeric value that we can later read, process and send to the Internet. Figure 1-4 shows how an analog sensor can be connected to a microcontroller device. All analog sensors usually come with 3 connection pins, one for receiving voltage (Vcc), one for ground connection (GND) and the output voltage pin (OUT).

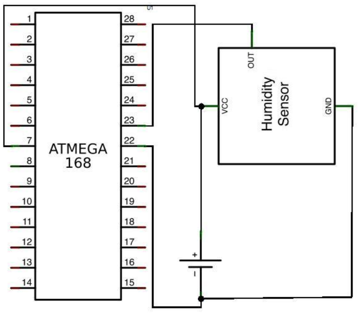

Figure 1-4. This is how a humidity sensor connects to the Atmega168 microcontroller. Vcc receives operating voltage and GND is connected to the ground of the power source. OUT is the pin that outputs the sensed signal and connects to microcontroller pin 23 (analog port).

Digital sensors generate what is called a Discrete Signal. Such a signal has different (usually binary only) states, like the logic gate states 0 and 1, which are represented by high and low current. For example, consider a push button switch (which by the way is one of the simplest forms of digital sensors): it has two discrete values. It is on, or it is off. Some more complicated sensors come also with a digital interface. This means that instead of generating a signal representing the voltage change between 0-5 based on their measurement, they actually generate a stream of bits (0 and 1 states).

More information on how sensors work is provided in chapter 2 of this book.

Communication Modules

The communication modules are parts of the device, which are responsible for the communication with the rest of the IoT platform. They provide connectivity according to the wireless or wired communication protocol they are designed for (such protocols are presented in the rest of this chapter). They usually consist of embedded electronic modules that implement the communication (i.e. transform the information received in bits and bytes to radio waves or signals that are transferred by wire respectively). Wireless modules usually consist also of an antenna for achieving maximum range, either external or internal for optimizing the size of the device.

Most likely the communication between IoT devices and the Internet is performed in two ways: a) there is an internet-enabled intermediate node acting as a gateway, b) the IoT device has direct communication to the Internet. In the first case, the device is commonly connected to the computer and sends data to it using e.g., a USB port. The computer receives the data and using appropriate software it forwards it to the Internet. In the second case, things are much simpler and devices can function and communicate more autonomously.

In either case, communication can be made according to the wireless technology used. Most modules that support wireless (like WiFi, Bluetooth and ZigBee) and wired technologies (like the Ethernet) also support the TCP/IP protocol. The TCP/IP protocol (for those unfamiliar with it) is the way computers, mobile phones and internet-enabled devices communicate with each other and to the Internet. It defines all communication essentials (e.g., how devices are identified by each other, using IP addresses; how information is split into small packets and transmitted; etc.). It also takes care of all connection setup issues and ensures information is properly delivered by retransmitting data whenever this is considered necessary.

What about inter-device communication? The communication between the main control units and the various communication modules is performed in most cases (and in the case of our projects) using the serial protocol. Serial is a device communication protocol that is standard on almost every PC and electronic device. The concept of serial protocol is simple. Each device has two ports (or connectors), one for transmitting data, usually annotated as Tx, and one for receiving data, usually annotated as Rx. Also a ground and power connection are required and are provided by the power source. Each serial port sends and receives bytes of information one bit at a time. So, in order to communicate, both control units (i.e. microcontrollers) and communication modules share a pair of Rx/Tx ports, as in Figure 1-5.

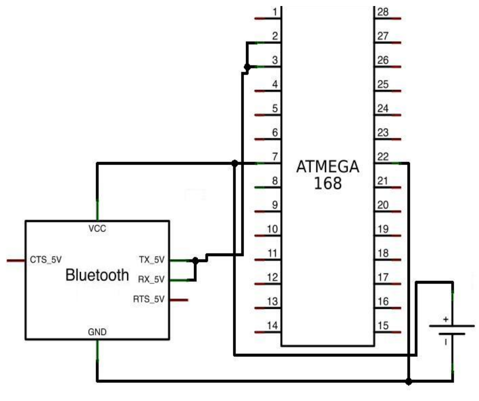

Figure 1-5. This is how a Bluetooth communication module connects to the Atmega168 microcontroller. Vcc receives operating voltage and GND is connected to the ground of the power source. Tx and Rx pins are the pins that send and receive serial data and connect to microcontroller pins 2 and 3 (Rx and Tx respectively).

Although this is slower than parallel communication, which allows the transmission of an entire byte at once, it is simpler and you can use it over longer distances. Because serial is asynchronous, the port can transmit data on one line while receiving data on another.

A brief presentation of communication enabling technologies for IoT devices follows in the ”Communication Technologies” section.

Power Sources

Every electronic device needs electric power to properly function. The electric power is the result of the potential difference between two points, otherwise known as voltage, and the electric current flowing through a circuit. In most electronic devices and also in the projects we are dealing with in this book, the two points are referred as the voltage input point (usually notated as Vin or Vcc) and the ground (usually notated as GND).

In small devices the current is usually produced by sources like batteries, thermocouples and solar cells. Batteries are electrochemical components that convert chemical power to electrical generating direct current (DC).

Wearable and mobile devices are mostly powered by small lightweight batteries that can also be recharged (like in Figure 1-6) for longer life duration. By utilizing simple circuits and appropriate programming of microcontrollers, devices can be aware of their battery status and alert users when they need to be recharged or replaced.

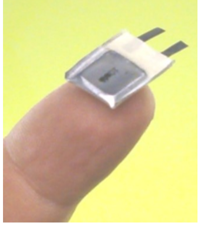

Figure 1-6. Probably the smallest rechargeable consumer battery. Weighs only 330mg and operates at 3.6V (image courtesy of PowerStream)

When selecting the appropriate battery source for your own projects, you look for the battery’s operational voltage and capacity. The latter is associated with the total energy stored within the battery. Capacity in general is measured in watt per hours (Wh), kilowatt per hours (kWh) or ampere per hours (Ahr). For sensors and other components of IoT projects you will build, the most common measure of battery capacity is Ah.

To give you an example about capacity and its actual meaning, if your IoT device needs 50mA to operate and your battery provides 50mAH, the latter will power your device adequately for about 1 hour before needing recharging.

Communication Technologies

‘Things’ need to talk to each other and also talk to the Internet in order to exchange sensor outputs, triggers, status messages etc. In order to do so, devices need to integrate a wireless (preferably) or a wired communication system. The major communication technologies that can be utilized by such devices are summarized in this section. In addition to the brief description of the technologies, samples of electronic modules that enable the respective communication are also presented. The modules selected are characterized by the special interfaces they have for connecting directly to microcontrollers and platforms like the ‘Arduino’ open-source electronics prototyping platform.

RFID

The Radio Frequency Identification (RFID) technology has been initially introduced for identifying and tracking objects with the help of small electronic chips, called tags. It is the most common technology behind asset tracking (that tells you where your mail parcel is before arriving its destination) and identifying objects (e.g., in automatic toll collection).

RFID tags are categorized into passive, active and battery assisted passive (BAP). The passive tags do not have a power source (battery) and thus cannot transmit and information on their own. They are powered-activated by the RFID reader and transmit only a small amount of information (usually an identification number (ID) of the tag). Active tags on the contrary have their own battery and can broadcast data continuously. A BAP tag can be considered as a hybrid: it carries a battery but only transmits information in presence of an RFID reader. The battery helps them to transmit their signal in longer distance than the passive tags (restricted to a few cm).

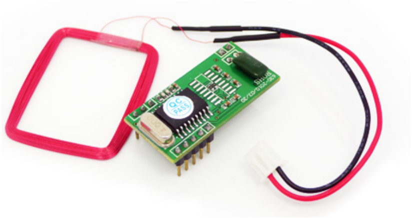

Figure 1-7. RFID module and its antenna. It can directly communicate with the Arduino using the Serial protocol. (image courtesy of Seeedstudio).

Most RFID tags consist of an integrated circuit for storing and processing the information, the essential radio frequency (RF) components for transmitting the information wirelessly and an antenna.

RFID has been initially characterized as the enabling communication power for the Internet of Things, due to its low cost, high mobility and efficiency in identifying devices and objects. Figure 1-7 illustrates an RFID module and its antenna that can directly communicate with a microcontroller.

Despite RFID being very popular for device identification and some information exchange (e.g., RFID-based temperature sensors) it cannot alone support the creation of IoT networks since it cannot provide any direct or indirect (e.g., through a gateway) communication to the Internet. The device proximity is also another drawback.

Bluetooth

You are using the Bluetooth technology to connect your wireless headset with your mobile phone, or to transfer photos from the latter to your computer. Bluetooth is a technology standard for exchanging data over short distances (using short wavelength radio transmissions in the ISM band from 2400-2480 MHz) from fixed and mobile devices, creating personal area networks. Bluetooth has been one of the first wireless communication protocols designed with lower power consumption for replacing short-range wired communications (in computer peripherals, mobile phone accessories, etc.). One important feature of Bluetooth is that devices can discover and communicate with each other without the need to be in visual line of sight (like in infrared communication), which is very important when using Bluetooth as a network technology for sensor systems deployment.

Bluetooth is named after the 10th century Danish King Harald Blåtand, which translates as Harold Bluetooth in English!

It is commonly used for connecting small devices with each other, due to the fact that it can support automatically the creation of peer networks (i.e. networks of devices that exchange and forward information) and provides communication functionality with low power consumption. The latter is very important for the case of IoT since many of the devices that one would like to interconnect to the IoT (sensors, actuators, etc.) have limited power resources. However on major drawback of Bluetooth is that it cannot provide direct connectivity to the Internet. One has to provide an intermediate node, e.g., a PC that will act as a gateway to the outer world. You can see how a Bluetooth communication module looks like in Figure 1-8.

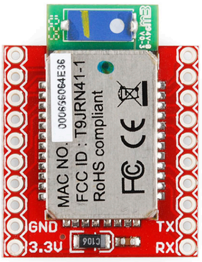

Figure 1-8. Bluetooth communication module on a breakout board for easily interfacing with devices like the Arduino. The module is so small it fits in your mobile phone, your hands free speaker and a USB dongle (image courtesy of Sparkfun).

ZigBee

ZigBee is one of the latest and most advanced wireless technologies being widely integrated into home automation & smart devices worldwide. It has been specifically developed as an open global standard to address the unique needs of low-cost, low-power wireless networks for communication between devices (also known as machine-to-machine or M2M networks). The ZigBee standard operates in unlicensed bands including 2.4 GHz, 900 MHz and 868 MHz

A ZigBee module (see Figure 1-9) can have a low consumption of 50mA. For instance, a rechargeable battery of 850mAH can provide about 17 hours of continuous operation for such module. Maximum data rate is about 250kbps and communication range can vary from 100m to 1km (maximum) depending on the output power (in small projects we usually use 1mW modules that provide maximum of 100m range).

Compared to Bluetooth, ZigBee provides better power efficiency, and higher range, making it thus a better wireless technology to consider for your IoT network. It also allows the automatic creation of peer networks and as a matter of fact it is consider being more extensible in this context.

However it still requires a gateway with Internet connectivity (e.g., a laptop) that will forward information from the Internet to the ZigBee network and vice versa.

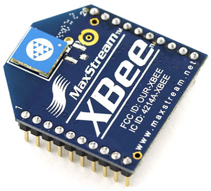

Figure 1-9. ZigBee communication module in a pluggable form known as XBee series modules (image courtesy of Sparkfun).

WiFi

WiFi, also known as the IEEE 802.11x standard, is the most common way to connect devices wirelessly to the Internet. Your laptop, smartphone and Tablet PC are equipped with WiFi interfaces and talk to your wireless router and provide you this way access to the Internet.

The commercially available WiFi modules (like the one in Figure 1-10) can be directly integrated to an IoT device and provide instant connectivity. The major advantage over the other wireless technologies is the fact that WiFi networks are very easy to establish and thus IoT devices with WiFi modules can have direct connection to the Internet. One drawback is the fact that this technology (which was at no means designed for IoT networks) is more power demanding than the others.

Figure 1-10. WiFi communication module in a pluggable form known as XBee series modules (image courtesy of Seeedstudio).

RF Links

Another option to connect devices and make them talk is utilize simple radio frequency (RF) interfaces. The latter are quite cheap and small (ideal when size matters) and can provide communication range between 100m and 1km (depending on the transmission power and the antenna used).

RF communication modules (see Figure 1-11) are connected to microcontroller and devices via through serial ports as the rest of the modules we have examined. However they do not provide any implementation of the TCP/IP communication protocol (or any other protocol). This means that if you want your devices to communicate you have to create your own protocol for establishing communication, identifying devices with each other and make sure all the information you have transmitted is delivered. Data rates are quite low (up to 1Mpbs) and you also need an Internet-enabled gateway that will provide access to your devices for making a complete IoT network.

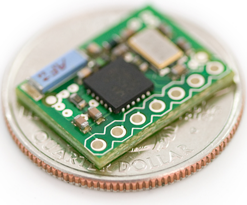

Figure 1-11. An RF transceiver module (it can be used both as a receiver and transmitter). It has low power consumption, can support data rates up to 1Mbps and can be easily interfaced to your Arduino (image courtesy of Sparkfun).

Cellular Networks: The Mobile Internet

The ‘Mobile Internet' refers usually to access to the Internet from a mobile device, such as a smartphone or laptop through a mobile broadband network. The mobile broadband network is based on cellular communication, same technology that is used in our mobile phones for serving our calls and text messages. It can provide direct Internet connectivity to a variety of data rates. Various network standards have existed for serving mobile Internet. GPRS, 3G, WiMax, and LTE (one of the 4G technologies) are a few to name. Depending on the standard and the available network coverage, connection speeds can go from 80Kbps (GPRS) to a few Mbps (3G and 4G).

Due to the complexity of the communication protocol and the information coding, in addition to high power requirements in cases where reception signal is low, the battery consumption of mobile Internet – enabled devices is an issue. Think of how fast the battery of your cell phone is drained when you browse the Internet. Still it is a good option for connecting devices directly to the Internet, since small GPRS modules for the Arduino are available (see Figure 1-12) and connectivity does not require further infrastructure (e.g., Internet connected laptop like in case of ZigBee or Bluetooth).

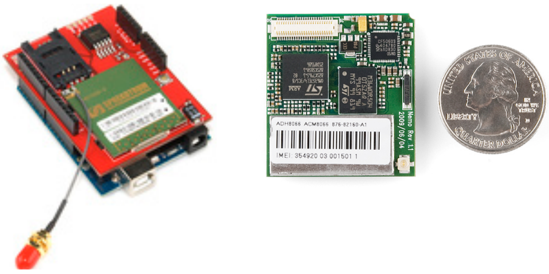

Figure 1-12. GPRS shield for your Arduino on the left. The GPRS module on the right. The small size of it allows it to fit easily in mobile phones, Tablet PCs and sensor devices (image courtesy of Sparkfun).

Wired Communication

Devices can definitely talk to each other and to the Internet using wired infrastructures, given the appropriate network interface. What is the best way? Since the very beginning of the Internet era to nowadays, your desktop computer has at least one Ethernet port.

The Ethernet protocol is very well established in computer communications. It does not require as much power as the wireless communications do, it can achieve very high data rates and most importantly it is so common in computer communications that all you need to do is plug an Ethernet cable to its device and get online. No worries about signal availability. It does not provide any mobility, but its advantages were the main reason it was one of the first networking technologies adopted in microcontroller platforms like the Arduino. Figure 1-13 presents an Ethernet module that can be directly plugged into the Arduino and an Ethernet-enabled Arduino board.

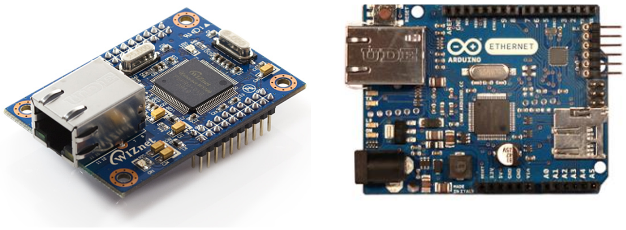

Figure 1-13. On the left: An Ethernet module. Can be directly connected to any sensor device and provide wired network functionality. On the right: An Arduino with embedded Ethernet interface (image courtesy of Arduino).

Which One Is the Best?

When coming to the point where we need to decide what is the most appropriate communication technology for your IoT network, there are several things we need to consider; mobility, network range, power consumption, size and cost are the most important to name. Table 1 – 1 provides an overview of the presented communication technologies comparing their features.

In case your devices will be at a fixed position, then an Ethernet module is one of the best options for providing communication at high-speed ranges with low power consumption. In a different case where mobility is a must (or there is no wired infrastructure available), we can select between WiFi, cellular and ZigBee. RF requires much more effort for building a communication protocol. ZigBee requires a gateway for providing connection to the Internet, but otherwise it is a very good solution, providing low power consumption and good coverage range. WiFi can provide direct access in case a WiFi infrastructure is available, but consumes much more power since it was originally designed for devices like laptops with power resources not for small IoT devices. When the only network infrastructure available is a mobile network, cellular data communication is the only option.

Table 1-1. Overview of the Wireless Communication Technologies.

Technology	Data rate	Range	Frequency
WiFi	54 Mbps	150 m	5 GHz
Bluetooth	721 Kbps	10 - 150 m	2.4 GHz ISM
RF – links	1 Mbps	50 – 100 m	2.4GHz ISM
IEEE 802.15.4, ZigBee	250 kbps, 20 kbps, 40 kbps	100 - 300 m	2.4 GHz ISM, 868 MHz, 915MHz ISM

Cellular 3G	14.4 Mbps/ 5.8 Mbps	m - km	800 MHz, 1900 MHz
Wired (Ethernet)	100 Mbps – 1 Gbps	m - km	-

Current Status and the Near Future

What has been the progress of the IoT since the initial description of the concept back in 1999? The evolution of the IoT in our daily lives depends mainly on two things; on the technical progress of microcontrollers and embedded devices and on the evolution of wireless interfaces in terms of communication and power efficiency.

Regarding the services for the IoT networks the current status involves several platforms that are already providing functionalities like hosting of data repositories, visualization of data, communication with the devices through web interfaces, remote triggering of events, etc. All these platforms and their services are specially designed and implemented for enabling the communication with IoT devices. This means that their functions are implemented using open and lightweight Internet web protocols that allow easy and direct communication, offering at the same time implementations in many programming languages and environments.

Examples of IoT services include the ones that will be presented in this book, like the Pachube data service and the Nimbits data logger and many more like the ThingSpeak and the iDigi Device Cloud. These services allow users to manage and visualize their sensor data either free or at low cost. Additional services like the recently discontinued Google Health or the Microsoft HealthVault collaborate with device vendors for managing device information on the web directly. Internet-enabled blood pressure monitors, glucose meters, pedometers, and similar health monitoring devices are available in the market enabling the user's to manage their health data from their web accounts.

The near future vision: It is already feasible to connect microcontroller platforms, like the Arduino, directly to IoT services and having them report sensor data and response to triggers set by users remotely through web applications. Embedded devices are already small enough to be even wearable, and advances in communications and power resources enable wireless Internet connectivity to all kinds of devices that we currently use. Imagine being able to control your home or office automation the way you view emails today; either from your mobile, your laptop or any place on the world with an Internet connection. Most importantly and much more interestingly, imagine that at the same time your automation devices will be able to retrieve important information for their function (e.g., weather updates) directly from the Internet make decisions, provide automations and require less input and interaction from you.

Explore the examples of this book and become ready to fully 'Internetize' your sensors and actuators by building your own IoT network with your devices!

Summary

This chapter has introduced you to the basic concepts of the Internet of Things. You have been provided with examples of IoT devices along with information about the major components (like microcontrollers and sensors). A brief presentation and comparison between the main communication technologies (like WiFi, Bluetooth, RF, Cellular, ZigBee and Ethernet) can give you ideas about how the IoT devices can interact with the Internet. The last section of this chapter has provided you information about the current status of the Internet of Things evolution and given you ideas about the near future.

The next chapter will explain you how sensors work and what kind of sensors you can utilize in your embedded projects for building your own IoT platforms.