**What Is a Sensor?**

Generally speaking, a sensor is a device that is able to detect changes in an environment. By itself, a sensor is useless, but when we use it in an electronic system, it plays a key role. A sensor is able to measure a physical phenomenon (like temperature, pressure, and so on) and transform it into an electric signal. These three features should be at the base of a good sensor:

* It should be sensitive to the phenomenon that it measures
* It should not be sensitive to other physical phenomena
* It should not modify the measured phenomenon during the measurement process

There is a wide range of sensors we can exploit to measure almost all the physical properties around us. A few common sensors that are widely adopted in everyday life include thermometers, pressure sensors, light sensors, accelerometers, gyroscopes, motion sensors, gas sensors and many more. A sensor can be described using several properties, the most important being:

* Range: The maximum and minimum values of the phenomenon that the sensor can measure.
* Sensitivity: The minimum change of the measured parameter that causes a detectable change in output signal.
* Resolution: The minimum change in the phenomenon that the sensor can detect.

**Sensor Classification**

Sensors can be grouped using several criteria:

* Passive or Active. Passive sensors do not require an external power source to monitor an environment, while Active sensors require such a source in order to work.
* Another classification is based on the method used to detect and measure the property (mechanical, chemical, etc.).
* Analog and Digital. Analog sensors produce an analog, or continuous, signal while digital sensors produce a discrete signal.

There are other ways and methods to group sensors but the classifications shown above are the easiest.

**How to Use Sensors in IoT**

The development of prototyping boards and the low price of sensors allow us easily use them in IoT projects. There are several prototyping boards on the market, suited for different projects depending on features and specifications. In this context, we will consider the two most popular boards: the Arduino Uno and Raspberry Pi 2.

This article will explore how to connect different sensors to these boards and how to interact with them.

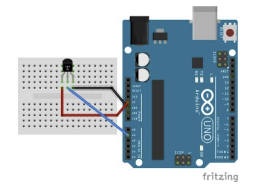
Before diving into the details on how to use sensors with these boards, it is important to note that every sensor has its own operating voltage range. This parameter is very important because the voltage supplied by the board must not be higher than the maximum voltage allowed by the sensor. Therefore, it is important to read the sensor data sheet carefully before connecting it to the board to avoid damage. The same principle is valid for the output signal, which must be lower than the maximum voltage that the board can tolerate.

**How to Use Arduino With Sensors**

The first and the most popular board is the Arduino Uno. It is a microcontroller board based on an ATmega328P. It is very easy to use, and a good starting point. This board provides 6 analog and 14 digital pins. It is perfect to use with analog and digital sensors.

**How to Measure the Temperature Using Arduino**

The easiest way to start is to connect an analog sensor to the Arduino. An analog sensor, as stated before, is a sensor that provides a continuous signal. For our first basic example, we will connect a simple temperature sensor, a TMP36. For more information, you can refer to the sensor data sheet. Generally speaking, the output voltage of this sensor is directly proportional to the environmental temperature. Arduino provides several analog input pins, labeled with an "A," that are suitable for accepting analog signals coming from a sensor. The schema below describes how to connect the sensor:



The code to read the temperature is very simple:

constinttempSensorPin = A1;

voidsetup() {

Serial.begin(9600);

}

voidloop() {

intpinValue = analogRead(tempSensorPin);

Serial.println("Pin value: " + String(pinValue));

float voltage = (pinValue / 1024.0) \* 5.0;

Serial.println("Voltage: " + String(voltage));

float temperature = (voltage - 0.5) \* 100; // °C

Serial.println("Temperature: " + String(temperature)

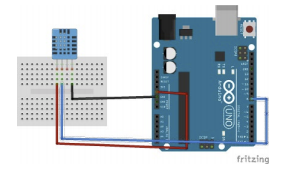
);

delay(5000);

}

**How to Measure the Temperature and Humidity Using Arduino**

Now it is time to connect a digital sensor to an Arduino. There are several digital sensors available, but for the sake of simplicity, we will consider a common digital sensor called the DHT11. This sensor measures the temperature and humidity. It is a very cheap sensor that provides a digital output. In this scenario, the sensor data pin has to be connected to the digital Arduino pin, as shown below:



The code is very simple. Even though we can parse the digital signal and read the temperature and the humidity, we will use a library to simplify development. The library is available in the Arduino IDE in the Sketch->Include library menu item.

#include "DHT.h"

#define PIN 8

#define DHTTYPE DHT11 // sensor type

DHT dht(PIN, DHTTYPE);

voidsetup() {

Serial.begin(9600);

}

voidloop() {

int temp = dht.readTemperature();

int hum = dht.readHumidity();

Serial.println("Temperature: " + String(temp));

Serial.println("Humidity: " + String(hum));

delay(5000);

}

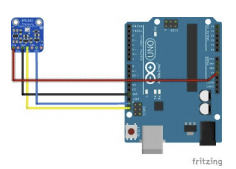
Running the code above, Arduino will log the temperature and the humidity every 5 seconds.

**How to Connect an I2C Sensor**

An I2C sensor is a serial bus used to connect peripherals to microprocessors. It is widely used and it requires four different pins:

* Vin
* GND
* CLK (Clock)
* SDA (Data)

To experiment with the I2C sensor with Arduino, we'll analyze the BMP280/BME280 sensor. This sensor measures, among other properties, the barometric pressure. The diagram below shows how to connect a BMP280 to Arduino:



As you can see, there are four different connections. The same connection can be used with a BME280. Do not forget to connect the sensor's CLK pin to the Arduino CLK and the SDA pin (the data) to the Arduino SDA. In addition, the SDO pin cannot be left floating, so you should connect it to the ground or to Vcc. The source code to read the pressure is shown below:

#include <Wire.h>

#include <Adafruit\_Sensor.h>

#include <Adafruit\_BMP280.h>

//BMP280

Adafruit\_BMP280 bmp;

voidsetup() {

Serial.begin(9600);

if (!bmp.begin()) {

Serial.println("Could not find a valid BMP280

sensor, check wiring!");

while (1);

}

}

void loop() {

float pressure = bmp.readPressure();

Serial.println("Pressure: " + String(pressure));

delay(5000);

}

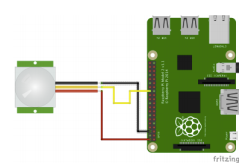
Before running the code above, you have to import a library to handle the sensor, as described in the previous example.

**How to Use Sensors With a Raspberry Pi**

Raspberry Pi is a single-board computer developed by the Raspberry Pi Foundation. There are several versions of Raspberry Pi with different specifications, but they all have their own operating system based on Linux. It is similar to a PC because it supports video output, USB ports, and keyboards. It is a very powerful board, and the examples below show only a little bit of its power.

**How to Use Movement Sensors**

To monitor movements, we will use a PIR sensor, which stands for Passive Infrared. It uses an infrared sensor to detect low-level radiation emitted by a warm body. Put simply, when the radiation level changes, it means a warm body is moving into its detection area. This sensor uses a digital pin that gets low (or high) when motion is detected. The schema below shows how to connect the sensor to Raspberry Pi. The connection may change if you use a different PIR version or a different Raspberry Pi board:



The Python code is shown below:

importRPi.GPIOas GPIO

import time

GPIO.setmode(GPIO.BCM)

sensorPin = 7

GPIO.setup(sensorPin, GPIO.IN)

whileTrue:

ifGPIO.input(sensorPIN) == GPIO.LOW:

print"Motion detected"

else:

print"No motion"

time.sleep(0.5)

When the PIR detects a movement, this simple app will log "Motion detected."

**How to Detect Gas**

Another interesting sensor is the MQ-4 sensor. The MQ-4 has a high sensitivity to natural gas. It can respond quickly, and is very easy to use. The connections between the sensor and Raspberry Pi are the same as in the PIR example. Be sure to use the sensor's digital pin, and be sure of the output voltage, which must be lower than 3V. If the sensor has an output greater than 3V, you have to use a logic-level converter. The code to use the MQ-4 is the same as the previous example.

**Summary**

At the end of this tutorial, we have learned how to use different types of sensors and how to connect them to the two most popular prototyping boards. It is possible to reuse the same sensors with different boards. Once you know how to read data from sensors, you have endless possibilities, and you're ready to explore IoT. You can use data from sensors by storing it in the cloud and using it later to create dashboards.

Sensors are everywhere. They’re in our homes and workplaces, our shopping centers and hospitals. They’re embedded in smart phones and an integral part of the Internet of Things (IoT). Sensors have been around for a long time. The first thermostat was introduced in the late 1880s and infrared sensors have been around since the late 1940s. The IoT and its counterpart, the [Industrial Internet of Things (IIoT)](https://en.wikipedia.org/wiki/Industrial_Internet_of_Things), are bringing sensor usage to a new level.

Broadly speaking, sensors are devices that detect and respond to changes in an environment. Inputs can come from a variety of sources such as light, temperature, motion and pressure. Sensors output valuable information and if they are connected to a network, they can share data with other connected devices and management systems.

Sensors are crucial to the operation of many of today’s businesses. They can warn you of potential problems before they become big problems, allowing businesses to perform predictive maintenance and avoid costly downtime. The data from sensors can also be analyzed for trends allowing business owners to gain insight into crucial trends and make informed evidence-based decisions.

Sensors come in many shapes and sizes. Some are purpose-built containing many built-in individual sensors, allowing you to monitor and measure many sources of data. [In brownfield environments](https://behrtechnologies.com/blog/from-brownfield-to-digital-factory-3-ways-to-iot-enable-your-legacy-systems/?__hstc=248538648.4f6db378c21791ec0222920b1f0db0db.1594315032149.1594315032149.1594315032149.1&__hssc=248538648.1.1594315032150&__hsfp=898855238), it’s key for sensors to include digital and analog inputs so that they can read data from legacy sensors.

There are many types of IoT sensors and an even greater number of applications and use cases. Here are 10 of the more popular types of IoT sensors and some of their use cases.

[IoT sensors have become critical to improving operational efficiency, reducing costs and enhancing worker safety.CLICK TO TWEET](https://twitter.com/intent/tweet?url=https%3A%2F%2Fbehrtech.com%2Fblog%2Ftop-10-iot-sensor-types%2F&text=IoT%C2%A0sensors%20have%20become%20critical%20to%20improving%20operational%20efficiency%2C%20reducing%20costs%20and%20enhancing%20worker%20safety.&via=behrtech&related=behrtech)

### 1. Temperature Sensors

Temperature sensors measure the amount of heat energy in a source, allowing them to detect temperature changes and convert these changes to data. Machinery used in [manufacturing](https://behrtechnologies.com/solutions/manufacturing/?__hstc=248538648.4f6db378c21791ec0222920b1f0db0db.1594315032149.1594315032149.1594315032149.1&__hssc=248538648.1.1594315032150&__hsfp=898855238) often requires environmental and device temperatures to be at specific levels. Similarly, within agriculture, soil temperature is a key factor for crop growth.



### 2. Humidity Sensors

These types of sensors measure the amount of water vapor in the atmosphere of air or other gases. Humidity sensors are commonly found in heating, vents and air conditioning (HVAC) systems in both industrial and residential domains. They can be found in many other areas including hospitals, and meteorology stations to report and predict weather.



### 3. Pressure Sensors

A pressure sensor senses changes in gases and liquids. When the pressure changes, the sensor detects these changes, and communicates them to connected systems. Common use cases include leak testing which can be a result of decay. Pressure sensors are also useful in the manufacturing of water systems as it is easy to detect fluctuations or drops in pressure.



### 4. Proximity Sensors

Proximity sensors are used for non-contact detection of objects near the sensor. These types of sensors often emit electromagnetic fields or beams of radiation such as infrared. Proximity sensors have some interesting use cases. In retail, a proximity sensor can detect the motion between a customer and a product in which he or she is interested. The user can be notified of any discounts or special offers of products located near the sensor. Proximity sensors are also used in the parking lots of malls, stadiums and airports to indicate parking availability. They can also be used on the assembly lines of chemical, food and many other types of industries.



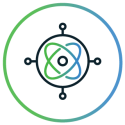
### 5. Level Sensors

Level sensors are used to detect the level of substances including liquids, powders and granular materials. Many industries including oil manufacturing, water treatment and beverage and food manufacturing factories use level sensors. Waste management systems provide a common use case as level sensors can detect the level of waste in a garbage can or dumpster.



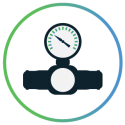
### 6. Accelerometers

Accelerometers detect an object’s acceleration i.e. the rate of change of the object’s velocity with respect to time. Accelerometers can also detect changes to gravity. Use cases for accelerometers include smart pedometers and monitoring driving fleets. They can also be used as anti-theft protection alerting the system if an object that should be stationary is moved.



### 7. Gyroscope

Gyroscope sensors measure the angular rate or velocity, often defined as a measurement of speed and rotation around an axis. Use cases include automotive, such as car navigation and electronic stability control (anti-skid) systems. Additional use cases include motion sensing for video games, and camera-shake detection systems.



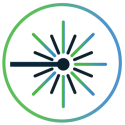
### 8. Gas Sensors

These types of sensors monitor and detect changes in air quality, including the presence of toxic, combustible or hazardous gasses. Industries using gas sensors include mining, oil and gas, chemical research andmanufacturing. A common consumer use case is the familiar carbon dioxide detectors used in many homes.



### 9. Infrared Sensors

These types of sensors sense characteristics in their surroundings by either emitting or detecting infrared radiation. They can also measure the heat emitted by objects. Infrared sensors are used in a variety of different IoT projects including healthcare as they simplify the monitoring of blood flow and blood pressure. Televisions use infrared sensors to interpret the signals sent from a remote control. Another interesting application is that of art historians using infrared sensors to see hidden layers in paintings to help determine whether a work of art is original or fake or has been altered by a restoration process.



### 10. Optical Sensors

Optical sensors convert rays of light into electrical signals. There are many applications and use cases for optical sensors. In the auto industry, vehicles use optical sensors to recognize signs, obstacles, and other things that a driver would notice when driving or parking. Optical sensors play a big role in the development of driverless cars. Optical sensors are very common in smart phones. For example, ambient light sensors can extend battery life. Optical sensors are also used in the biomedical field including breath analysis and heart-rate monitors.



### MYTHINGS IoT Sensor

The MYTHINGS Smart Sensor is a self-contained, battery-powered multi-purpose IoT sensor that allows you to capture critical data points like acceleration, temperature, humidity, pressure and GPS. The smart sensor is integrated with the MYTHINGS Library – a hardware independent, small-footprint and power-optimized library of code, featuring the MIOTY (TS-UNB) low-power wide area network protocol.

The Internet of Things is rarely discussed without the conversation steering to data and the new Data Economy. The intelligence and value from an IoT system is based on what can be learned from the data. Sensors are the source of IoT data.

Driven by new innovations in materials and nanotechnology, sensor technology is developing at a never before seen pace, with a result of increased accuracy, decreased size and cost, and the ability to measure or detect things that weren’t previously possible. In fact, sensing technology is developing so rapidly and becoming so advanced that we will see a trillion new sensors deployed annually within a few years.

## Sensors and Actuators

A better term for a sensor is a transducer. A transducer is any physical device that converts one form of energy into another. So, in the case of a sensor, the transducer converts some physical phenomenon into an electrical impulse that can then be interpreted to determine a reading. A microphone is a sensor that takes vibrational energy (sound waves), and converts it to electrical energy in a useful way for other components in the system to correlate back to the original sound.

Another type of transducer that you will encounter in many IoT systems is an actuator. In simple terms, an actuator operates in the reverse direction of a sensor. It takes an electrical input and turns it into physical action. For instance, an electric motor, a hydraulic system, and a pneumatic system are all different types of actuators.

In a typical IoT system, a sensor may collect information and route to a control center where a decision is made and a corresponding command is sent back to an actuator in response to that sensed input. Later, we will discuss where the control center resides in the greater IoT system.

There are many different types of sensors in an IoT system. Flow sensors, temperature sensors, voltage sensors, humidity sensors, and the list goes on. In addition, there are multiple ways to measure the same thing. For instance, airflow might be measured by using a small propeller like the one you would see on a weather station. Alternatively, as in a vehicle measuring the air through the engine, airflow is measured by heating a small element and measuring the rate at which the element is cooling.

Different applications call for different ways of measuring the same thing.

### The Importance of Accurate Sensors

Imagine that you are a bar owner and you want to measure the amount of beer coming out of one of your taps. One way you might do this is to install a sensor in line with the line that runs from the keg of beer to the tap. This sensor would most likely have a small impeller inside of it. When the beer ran through the sensor, it would cause the impeller to spin, just like the propeller on a weather station.

When the impeller spins, it will send a stream of electrical impulses to a computer. The computer will interpret the impulses to determine how much beer is flowing through. Sounds simple, right?

This is where sensors get interesting. If you look back at our description, you’ll see that we never directly measured the amount of beer flowing through the sensor; we interpreted it from a stream of electrical impulses. That means that we must first figure out how to interpret it. Calibration.

To calibrate the sensor, we’d have to take a container with a known carrying capacity, say, a pint glass. Then we’d have to fill that container under a variety of conditions to determine what the electrical pulse signal looked like.

For instance, the first pour off a new keg might tend to have more foam, which would read differently than a pour from the middle of the keg that was all beer. It’s only through repeated trials and a lot of data that we gain confidence that we can interpret the data and determine how much beer was poured.

Once the correlation is well known, a protocol can be developed to always assure the sensor is reading correctly. This is called calibration. Reputable manufacturers will deliver fully calibrated devices and provide instruction on how to re-calibrate to verify sensor accuracy.

The accuracy of sensed data is paramount, since you will make mission-critical decisions based on later analysis of the data, which will hold little value if the data is wrong.

Read more about the IoT Technical Stack in “[A Reference Guide to IoT](https://bridgera.com/ebook/)” eBook.

### Important Next Steps

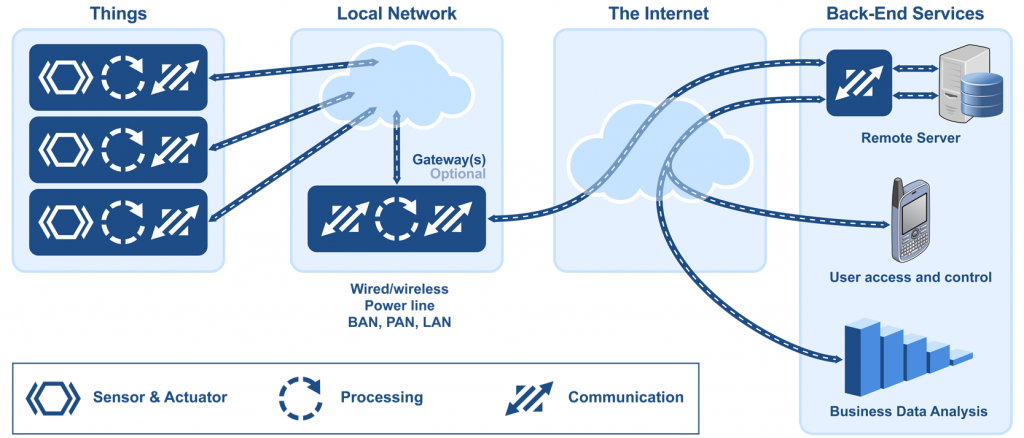
Once you have your device engineering completed, you will be ready to prototype your IoT System which is most effective with an IoT cloud solution that can ingest device data and provide a way to query or visualize the data. [BridgeraIoT](https://bridgera.com/) is a cloud platform and a service that allows companies to produce something quickly that works, without the compromise of a poorly built back end or generic UI, therefore enabling a more meaningful demonstration and supporting a more effective test/validation effort. Then, most importantly, converting right into a highly scalable model when ready.

## How to Think about the Internet of Things (IoT)

Many people have tried to define the Internet of Things. But as a hardware or software engineer, you already know the essential element: to build interconnected products.

In broad strokes, there are four main components of an IoT system:

* The Thing itself (that is, the device)
* The local network (this can include a gateway, which translates proprietary communication protocols to Internet Protocol)
* The Internet
* Back-end services (enterprise data systems, or PCs and mobile devices)

[](https://www.micrium.com/wp-content/uploads/2014/03/internet-of-things.png)

The Internet of Things from an embedded systems point of view

IoT systems are not complicated, but designing and building them can be a complex task. And even though new hardware and software is being developed for IoT systems, we already have all the tools we need today to start making the IoT a reality.

We can separate the Internet of Things into two broad categories:

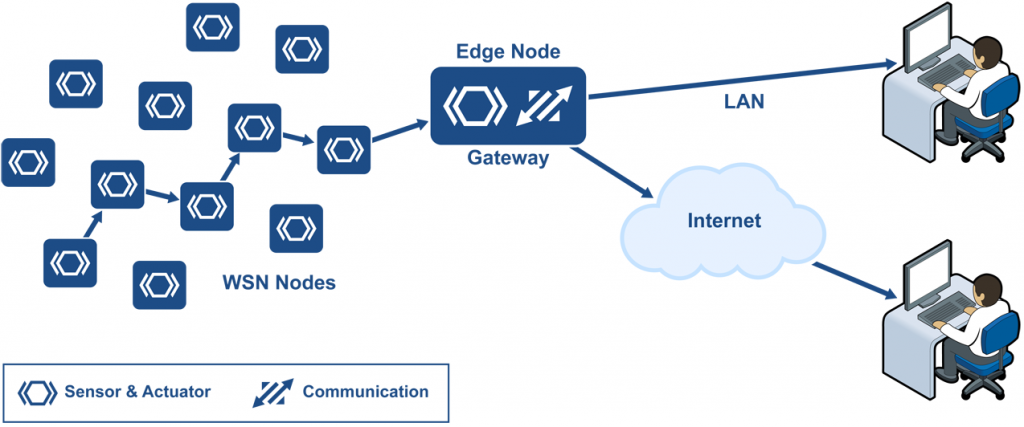
* **Industrial IoT**, where the local network is based on any one of many different technologies. The IoT device will typically transmit data over the global Internet.
* **Commercial IoT**, where local communication is typically either Bluetooth or Ethernet (wired or wireless). The IoT device will typically communicate only with local devices.

So, to better understand how to build IoT devices, you first need to figure out how they will communicate with the rest of the world.

## Your Local Network

Your choice of communication technology directly affects your device’s hardware requirements and costs. And IoT devices are deployed in so many different ways — in clothing, houses, buildings, campuses, factories, and even in your body — that no single networking technology can fit all bills.

Let’s take a factory as a typical case for an IoT system. A factory would need a large number of connected sensors and actuators scattered over a wide area, and a wireless technology would be the best fit.

[](https://www.micrium.com/wp-content/uploads/2014/03/wireless-sensor-network.png)

Wireless sensor network installed in a factory, connected to the Internet via a gateway

A wireless sensor network (WSN) is a collection of distributed sensors that monitor physical or environmental conditions, such as temperature, sound, and pressure. Data from each sensor passes through the network node-to-node.

## WSN Nodes

WSN nodes are low-cost devices, so they can be deployed in high volume. They also operate at low power so that they can run on battery, or even use energy harvesting. A WSN node is an embedded system that typically performs a single function (such as measuring temperature or pressure, or turning on a light or a motor).

Energy harvesting is a new technology that derives energy from external sources (for example, solar power, thermal energy, wind energy, electromagnetic radiation, kinetic energy, and more). The energy is captured and stored for use by small, low-power wireless autonomous devices, like the nodes on a WSN.

## WSN Edge Nodes

A WSN edge node is a WSN node that includes Internet Protocol connectivity. It acts as a gateway between the WSN and the IP network. It can also perform local processing, provide local storage, and feature a user interface.

## WSN Technologies

The battle over the preferred networking protocol is far from over. There are multiple candidates.

### Wi-Fi

The first obvious networking technology candidate for an IoT device is Wi-Fi, because it is so ubiquitous. Certainly, Wi-Fi can be a good solution for many applications. Almost every house that has an Internet connection has a Wi-Fi router.

However, Wi-Fi needs a fair amount of power. There are myriad devices that can’t afford that level of power: battery operated devices, for example, or sensors positioned in locations that are difficult to power from the grid.

### Low-Power Solutions

The newest networking technologies allow for low-cost, low-power solutions. These technologies support the creation of very large networks of very small intelligent devices.

Currently, major R&D efforts include:

* Low-power and efficient radios, allowing several years of battery life
* Energy harvesting as a power source for IoT devices
* Mesh networking for unattended long-term operation without human intervention (for example, M2M networks)
* New application protocols and data formats that enable autonomous operation

For example, [EnOcean](http://www.enocean.com/en/home/) has patented an energy-harvesting wireless technology to meet the power consumption challenge. EnOcean’s wireless transmitters work in the frequencies of 868 MHz for Europe and 315 MHz for North America. The transmission range is up to 30 meters in buildings and up to 300 meters outdoors.

### IEEE 802.15.4

One of the major IoT enablers is the IEEE 802.15.4 radio standard, released in 2003. Commercial radios meeting this standard provide the basis for low-power systems. This IEEE standard was extended and improved in 2006 and 2011 with the 15.4e and 15.4g amendments. Power consumption of commercial RF devices is now cut in half compared to only a few years ago, and we are expecting another 50% reduction with the next generation of devices.

### 6LoWPAN

Devices that take advantage of energy-harvesting must perform their tasks in the shortest time possible, which means that their transmitted messages must be as small as possible. This requirement has implications for protocol design. And it is one of the reasons why 6LoWPAN (short for IPv6 over Low power Wireless Personal Area Networks) has been adopted by ARM (Sensinode) and Cisco (ArchRock). 6LoWPAN provides encapsulation and header compression mechanisms that allow for briefer transmission times.

| Wireless radio technologies | | | |
| --- | --- | --- | --- |
| **Standard** | **IEE 802.15.4** | **Bluetooth** | **Wi-Fi** |
| Frequency | 868/915 MHZ, 2.4 GHz | 2.4 GHz | 2.4, 5.8 GHz |
| Data rate | 250 Kpbs | 723 Kpbs | 11 to 105 Mpbs |
| Range | 10 to 300 m | 10 m | 10 to 100 m |
| Power | Very Low | Low | High |
| Battery Operation | Alkaline (months to years) | Rechargeable (days to weeks) | Rechargeable (hours) |

There are many wireless networks available that are specialized for various industries. The following is a brief list:

|  |  |  |
| --- | --- | --- |
| 6LoWPAN | DASH7 | Wireless M-Bus |
| ANT | ISA100 | Z-Wave |
| Bluetooth | Wireless HART | Zigbee and Zigbee IP |

And there are many more.

At Micrium, we believe that any protocol that carries IP packets has an advantage over all others. The connectivity requirements for IoT devices are so diverse that a single technology cannot meet all the range, power, size and cost requirements. Nonetheless, we believe that 6LoWPAN will be the choice for WSNs and light IP-based protocols (see next section).

## IPv6 is Key for IoT

If your IoT network is local and M2M-only, then the wireless protocols discussed above are all good candidates. But if your goal is to remotely control devices or otherwise transmit data over the Internet, then you need IPv6.

The usefulness of IoT devices resides not only in local communication, but also in global communication. If at all possible, it is crucial that your IoT networks (LANs, PANs, and BANs) all make use of the suite of Internet Protocols (IP, UDP, TCP, SSL, HTTP, and so on). A stable server is also required – [ITT Systems has a list of free Servers you can download](http://www.ittsystems.com/best-free-tftp-servers-windows/) – which will ensure your project runs smoothly across all platforms. Furthermore, your networks must support [Internet Protocol version 6](https://www.micrium.com/ipv6/), as the current IPv4 standard faces a global addressing shortage, as well as limited support for multicast, and poor global mobility.

IPv6’s addressing scheme provides more addresses than there are grains of sand on earth — some have calculated that it could be as high as 1030 addresses per person (compare that number to the fact that there are 1028 atoms in a human body)! With IPv6, it is much simpler for an IoT device to obtain a global IP address, which enables efficient peer-to-peer communication.

The importance of IP to the Internet of Things does not automatically mean that non-IP networks are useless. It just means that non-IP networks require a gateway to reach the Internet.

Referring back to the illustration at the top of the page, you can see clearly that your local network is only one part of the Internet of Things. 6LowPAN, because it carries an IPv6 address with a compressed header, offers Internet connectivity without too much additional overhead. 6LoWPAN has also an advantage over other personal area networks, because peer-to-peer communication is simpler to implement when each device has a global address.