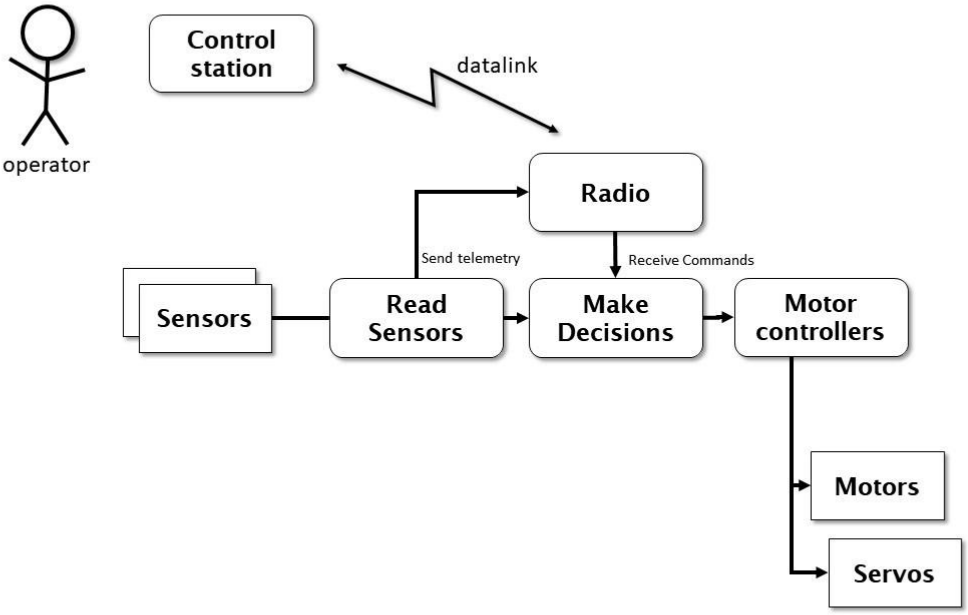
What is a robot?

The word robot entered the modern language from the play R.U.R. by the Czech author, Karel Capek, which was published back in 1920. Roboti is supposed to be a Czech word meaning forced servitude. In the play, an industrialist learns how to build artificial people – not mechanical, metal men, but made of flesh and blood, who emerge from a factoryfully grown. The English translation of the name as Rossum's Universal Robots (R.U.R.) introduced the word "robot" to the world.

Robot anatomy – what are robots made of?

In our limited definition of **robots** as meaning mobile machines that have sensors and interact with their environment, there is a fairly standard collection of components and parts that make up the vast majority of robots. Even robots as outwardly different as a self- driving car, the welding robot that built the car, and a Roomba vacuum cleaner, are actually composed of some version of the same parts. Some will have more components, others less, but pretty much all robots will have the following categories of parts

(*Reference: Murphy, Robin R.* ***Introduction to AI Robotics****, MIT Press, 2000, Chapter 1):*



- **Controller**: A unit that runs the programming that controls the robot. This can be a traditional computer, a microcontroller, a **single board computer** (**SBC**), like the one we have, or some other sort of processor that sends and receives commands. Robot arms and some types of industrial robots will use a **programmable logic controller** (**PLC**), which mimics the way robots were programmed before digital computers with relays.
- **Control station** or **human/robot interface** (**HRI**): Robots are designed to perform tasks, which require that humans must have some means of sending instructions to the robot and supervising that the robot is behaving correctly. We will be using a laptop or desktop computer for this function, and will talk to the robot via a network. Control stations send commands to the robot, and receive **telemetry** from the robot in the form of **data**, **video**, or **audio**.
- **Radio** or **datalink**: Mobile robots, like the one we are designing in this book, are capable of moving and exploring their environment. While it is possible to send commands to a robot over a tether or wire, a preferred way is to use a radio link. The ubiquitous availability of wireless networks, such as Wi-Fi and cellular data services, has made creating data links a lot easier. I have had a lot of robot projects where a network link was unavailable or impractical, and a custom radio solution needed to be devised. Other types of radio used in robots may

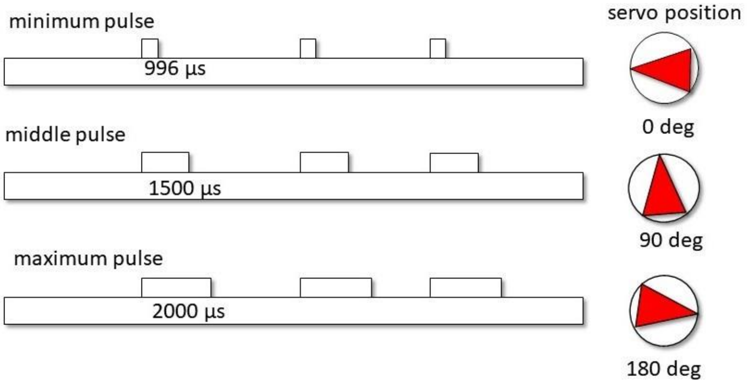
include Bluetooth, Zigbee, and various mesh network systems, such as Flutter.

- **Motors** or **effectors**: Our definition of a robot included the ability for self- propulsion – in other words, the robot is able to move. In order to move, the robot needs one or more motors. Our robot, TinMan, has eight motors, two for driving, and six for controlling the robot arm and hand. Motors, in general, convert electricity into motion. There are many, many different types, and picking the correct motor is difficult. You must match the torque (how hard the motor can pull), the speed in revolutions per minute, and voltage. Many robot motors also feature gearboxes to slow the speed of the motor down, basically exchanging speed for torque. TinMan’s electric motors have reduction gear boxes.

There are other ways to provide motion to a robot, in fact, many other ways. We call these things that make the robot move **effectors**. Effectors are only limited by your imagination, and include pneumatics (systems actuated by compressed air), hydraulics (things actuated by incompressible fluid), linear actuators that go back and forth rather than round and round, and even exotic effectors, such as Shape Memory Alloy or piezoelectric crystals, which change shape when electricity is applied. There are robot fish, robot octopi, squishy robots that fit under doors, robots that roll, robots that jump, and robots that walk.

- **Servos**: Some of the motors in our robot are a special category of motors called **Servos**. Servo motors feature a feedback mechanism and a control loop, so that they behave in a particular manner, either to maintain a position or a speed. The feedback is provided by a sensor. The servos we are using consist of a small electric motor that drives a gearbox made up of a whole series of gears that reduce the speed and increase the torque of the motor. The sensor is a potentiometer (variable resistor) that can measure the angle of the output gear shaft. When we send a command to the servo, it tells the motor to set to a particular angle. The angle is measured by the sensor, and any difference between the motor's position and the sensor creates an error signal that moves the motor in the correct direction. You can hear the motor making a lot of noise, because the motor turns many times through seven reduction gears to make the arm move. The gearbox lets us get a lot of torque without drawing a lot of current.

- **Motor controller**: Motors are not very useful by themselves – you need some capability to convert commands from the control computer into motion from the motors. Since motors require more voltage and current than the control computer (our Raspberry Pi) can provide, we need a device to turn small digital signals into large analog voltage and current. This device is called a **motor controller**, or sometimes an **electronic speed control**. Since we have a tank-drive robot, we also need the motors to be able to run in forward or reverse. The motor controller takes a special input signal called a **pulse width modulation** (**PWM**). PWM is a repeating signal where the voltage turns on and off. The motor throttle (how fast the motor turns) is proportional to the amount of time the PWM signal stays in the **ON** position:



The motor controller has several kinds of connections, and has to be wired carefully due to the higher voltages and currents provided. There are two control wire inputs – one for speed (the PWM signal), while the other is a direction signal. We put the motor in reverse by changing the direction signal – 1 is forward, and 0 is backward. The next thing we need is a ground – it is very important that the controller sending the PWM signal (in our case, the Arduino Mega) and the motor control have their ground lines connected together. We'll cover this in the *wiring* section later in the chapter. Next, the motor controller needs the motor voltage and current, which we get directly from our battery. Finally, we connect two wires from each motor to the controller. It is interesting that we don't care which wire goes to which side of the motor, since we can run both forward and backward. If the motor is turning the wrong way, just switch the two wires. This is the only time you get to say *just reverse the polarity* outside of a science fiction movie!

- **Sensors**: A robot can be described as a machine that can move and react to its environment. In order for the robot to be able to see its surroundings, it needs **sensors.** Sensors take information from the outside or inside of the robot and convert it into a digital or electronic format. If we use a camera sensor, it takes light information and turns it into digital pixels recorded as an array of numbers. A sonar sensor would measure the distance to an object, such as a wall, by sending a pulse of energy (sound waves) and listening for the time delay before hearing an echo. Measuring the time delay gives us the distance to an object. For our TinMan project, the robot has several types of sensor, some more obvious than others. Our primary sensor is a wide-angle video camera, which we will use for avoiding obstacles and detecting objects. We will also use a microphone to listen for sounds and perform speech recognition. We mentioned servo motors in a previous paragraph – each servo motor contains an angle sensor that detects the amount of rotation, and allows us to direct the robot arm and hand. We have our Emergency Stop button, which is wired to the Arduino, and is a type of tactile (touch) sensor. When the button is pressed, the digital signal line to the Arduino is connected to ground, which we can detect as a 0 value on that pin.The Arduino can also perform an **analog-to-digital** (**A2D**) conversion. This means we can put a voltage on an Arduino pin from 0 to 5 volts and measure the amount of that voltage. We could use that to monitor our power supply, or to measure temperature via a thermistor. A **thermistor** is just a device that changes resistance with temperature. If we put a voltage through the thermistor, we can measure changes in that voltage on the thermistor's output.