Chapter 2

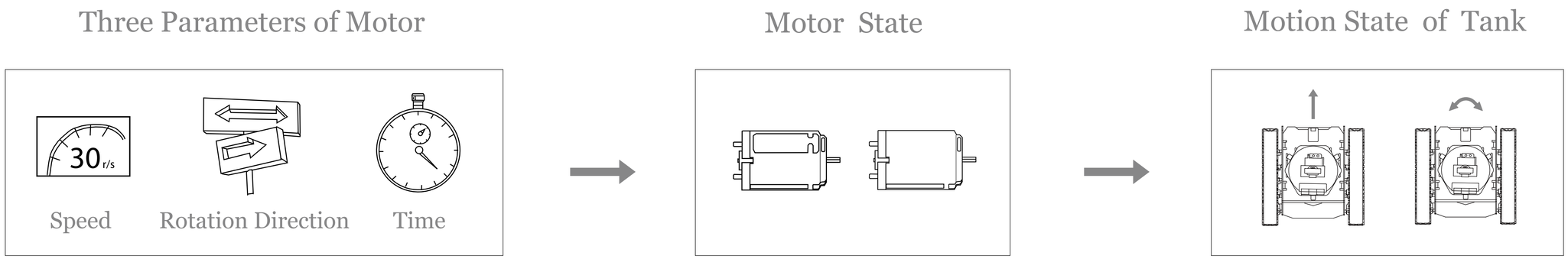
Run! Devastator!

In chapter 1, we went through the hardware assembling procedure, now it's the time to get it moving! In the following sessions, we will show you how to set up a simple motor driving program step by step.

Moving mechanism

The mobility of the Devastator is realized by the motions of its motors on each side. The motion of each motor follows the signal sent by the micro controller.
To be more specific, 3 parameters are included in a signal, each determines motor's angular speed, rotation direction and corresponding time duration.

The angular speed determines how fast the motor drives the pedrails
The rotation direction determines the moving direction of the pedrail
The time duration determines how long a certain motion lasts



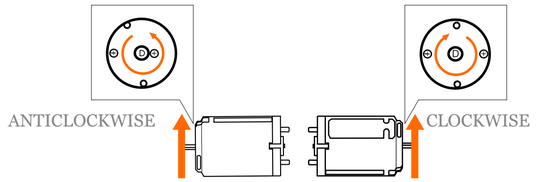
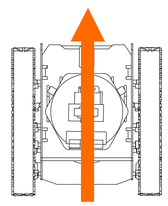
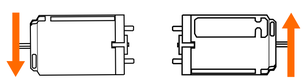
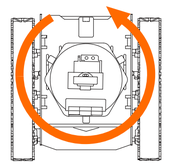
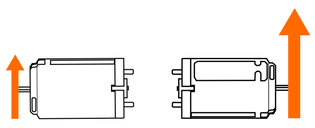
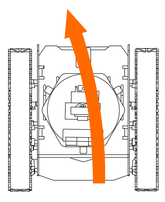
How to steer the Devastator

The steering angle is determined by the speed difference between its two pedrails.

When both pedrails are driven at the same speed, our robot may either go straight forward or backward.

However, when speed of these two motors are inconsistent, the larger the difference is, the bigger its steering angle will be.

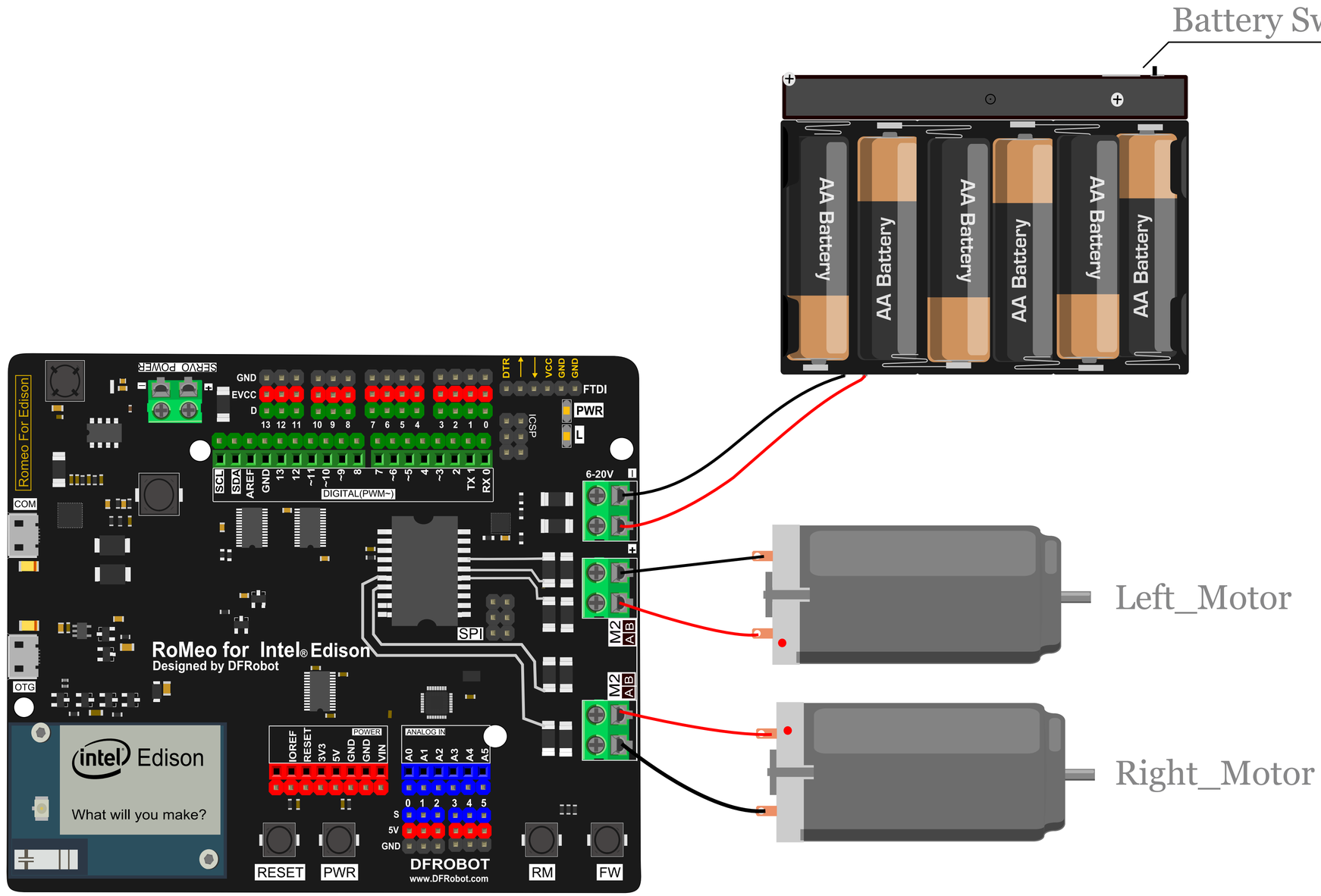
Note: as motors are installed to be oriented in opposite directions, an anticlockwise rotation of the left motor and a clockwise rotation of the right motor drives pedrails in the same direction.

 Same speed of Motor Pedrail moving in the same direction	 Go Straight
 Same speed of Motor Pedrail moving in the opposite direction	 Around in Circles
 Different speed of Motor	 Deflection Motion

Steering mechanism

Wire up components

As we have already finished the installation of the hardware part, all what we need to do is to check if the circuit connection is correct.



Circuit connection

Upload code

Before hitting the upload button, we need to compile the code. To get this done, libraries that we included in our code should be imported to our local Arduino software first.

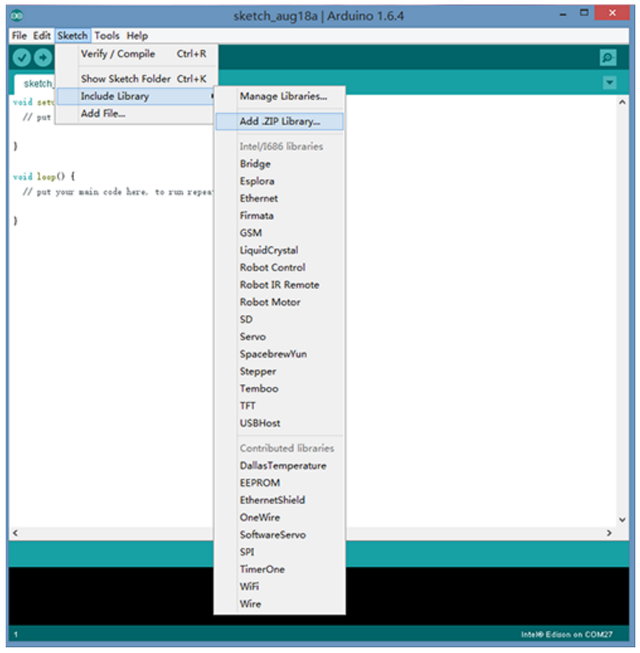
Import library

A simple way to understand how libraries work is to think them as third party owned arsenals. When certain types of weapons are needed, we don't have to build them by ourselves. Instead, all we need is to take the storage list as reference and borrow whatever we need that exists in the list. Back to the concept of library, similarly, once a library is included in our code, functions supported by this library can be utilized by simply calling a few statements, which saves a lot of time and space. For instance, if we want to control our motor, by calling statements from a motor control library, all we need is to enter these 3 parameters that we mentioned above in a preset format (we will go deep into this in the next session). Another advantage of using libraries is that we don't even have to understand how exactly these codes work, all we care is their functions and formats.

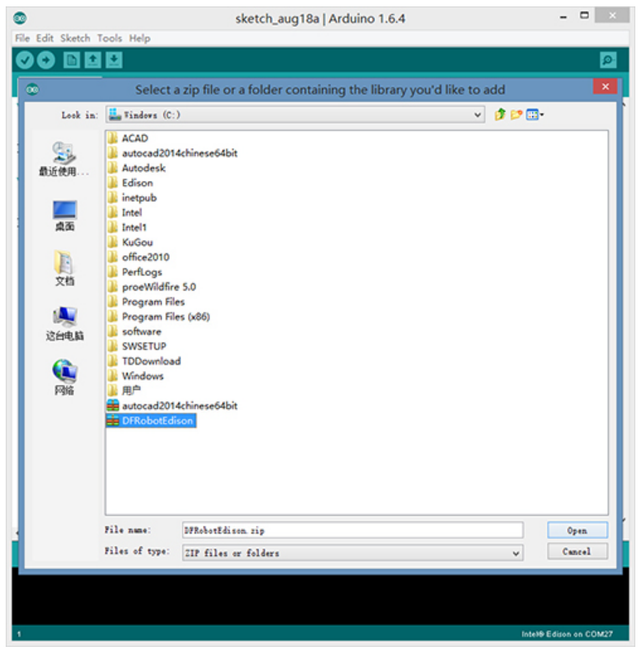
Alright! Let’s start importing libraries to the Arduino software.

Download the DFRobotEdison library in zip format from the link below

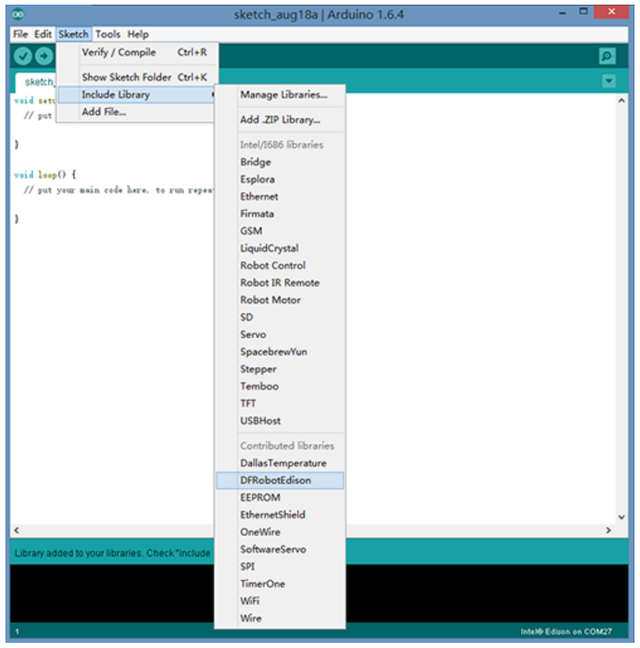
Import the library to our local Arduino software



Once download is complete, go to the menu bar of Arduino software and select Sketch > **Include Library** > **Add .ZIP Library**



Select the library file in zip format and click **Open**



There you go! Now the library has been imported into the Arduino software. To make sure it's correctly imported, you can also go to **Sketch** > **Include Library** and check if the name of the library is in the list.

Now it's time to code. Let's download a simple sample program first.

Upload testing code

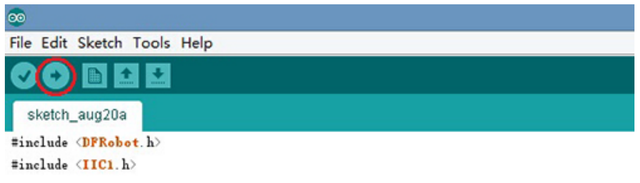
After all preparations are done, it's the time to do some programing! To begin with, open the Arduino IDE and enter following code.

Sample code

```
#include <DFRobot.h>
#include <IIC1.h>
 DFrobotEdison L_Motor;
DFrobotEdison R_Motor;
void setup() {
  L_Motor.begin(M2);    /*Initializes the motor drive*/
  R_Motor.begin(M1);
}
 void loop() {
  L_Motor.setDirection(CLOCKWISE);  /*Motor clockwise rotation*/
  R_Motor.setDirection(CLOCKWISE);
  L_Motor.setSpeed(255); /*Motor speed*/
  R_Motor.setSpeed(255);
  delay(2000);
  L_Motor.stop(); /*Stop*/
  R_Motor.stop();
  delay(3000);
}
```

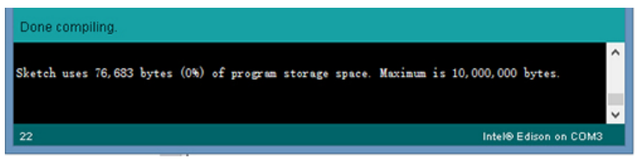
For convenience, you may simply copy and paste the code to Arduino IDE. However, for beginners, we recommend you to type the code manually so you would have a basic idea of the C language that we use to program our robot. Need to mention that it is crucially important to pay attention on every single character in the code. Any mistake would lead it fail to compile.

Before uploading the code, always remember to check that the correct board type (Intel® Edison) and serial port (COM # port) have been selected, then click Verify below the menu bar.



Compiling in progress

Done!



The next step is to click “**Upload**” right next to “**Verify**”, then we are all set!!

If everything goes smoothly, our robot should be doing the following movements: in every 5 seconds, it moves forward at its maximum speed for 2 seconds and then stops for 3 seconds.

Code Analysis

Include libraries

```
#include<DFRobot.h>
#include<IIC1.h>
```

In the beginning of the code, we need to include libraries by calling the #include statement so their functions can be utilized by our program.

Define motor name

```
DFrobotEdison L_Motor;
DFrobotEdison R_Motor;
```

“DFrobotEdison” is the type of the variable predefined in the library “DFRobot.h”. By defining the name L_Motor and R_Motor as “DFrobotEdison” variables, we can set the motor to do functions supported by the library.

Initialize motor drive

```
void setup() {
  L_Motor.begin(M1); /*Initializes the motor drive*/
  R_Motor.begin(M2);
}
```

Inside the setup function, use the XXX.begin() statement to pair up motors to motor driving ports.

Motion control

As L_Motor and R_Motor has been defined as “DFrobotEdison” variable, according to its library, we can enter parameters in following formats to control the motion of each motor.

Direction control

```
L_Motor.setDirection();
R_Motor.setDirection();
```

Enter CLOCKWISE or ANTICLOCKWISE into the bracket of in the statement above to make the motor either go clockwise or anticlockwise.

Speed control

```
L_Motor.setSpeed();
R_Motor.setSpeed();
```

Enter an integer from 0-255 into the bracket in the statement above to control the speed of the motor, while 255 corresponds to its maximum speed

Time duration

```
delay();
```

Enter an integer into the bracket in the above statement to set the time duration in milliseconds.

Examples

By applying the statements above, program the following movements

Movement 1:

Reverse the direction of the left motor to switch the robot from going straight forward to moving round in a cycle anticlockwise.

L_Motor.setDirection(CLOCKWISE); R_Motor.setDirection(CLOCKWISE);	L_Motor.setDirection(ANTICLOCKWISE); R_Motor.setDirection(CLOCKWISE);

Movement 2:

Lower the speed of the left motor to switch the robot from going straight forward to going left forward

L_Motor.setSpeed(255); R_Motor.setSpeed(255);	L_Motor.setSpeed(100); R_Motor.setSpeed(255);