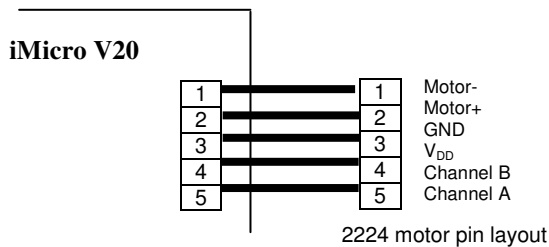


Application Note: How to Use DC Motor with iMicro Robotic Controller

The iMicro Controller Board can control 2 DC motors. The motors can be connected to J11 (Motor0) and J12 (Motot1) headers. The pin configurations of the header pins are as follow:

| | | | | | |
|----|----|-----|----|------|------|
| M- | M+ | GND | 5V | CH-B | CH-A |
| 1 | 2 | 3 | 4 | 5 | 6 |

For example, if the FAULHABER Minimotor 2224 with encoder is to be used with the iMicro board, the connection of the motor will be as follows.



Note: Connecting the pins wrongly may results in damage of incremental encoder.

After connecting the motors, you can switch on the power and the motors must be in locked position. If they are running very fast or can be moved freely, check the connection to the motors. It is common to see this kind of problem when Motor- and Motor+ pins are plugged in wrongly. And it is advisable to pay extra attention in plugging motors as the encoder of the motor can be destroyed instantly if the V_{DD} and GND are connected wrongly.

When the motors are connected properly and they are in locked position, the motion control can be done easily by using the provided function library. For full list of motion control functions, please refer to [iMicroV11 Software Guide](#).

For example, the velocities of the motors can be set as simple as the code below.

```
float pV0,pV1;
pV0 = 1000;
pV1 = 1000;
SetVelocity(&pV0, &pV1);
```

Note that the unit of the velocity is millimeter per second (mm/sec) provided that the wheels are properly configured. For example, if the wheel of motor 0 in the robot is as follows,

Motor Gear Ratio: 1:10
 Wheel Diameter: 62mm
 Encoder CPR: 512 (note: for encoder with 2 channel output, the actual CPR is 4 times)

Use the following code to configure the wheel ONCE before calling any motion control function.

```
float pW,pG, pE;
pG = 10.0;
pW = 62.0;
PE = 2048.0; //4 times of the CPR of the encoder disk
SetWheel0Info(&pG, &pW, &pE);
```

To read back the current velocity of the motor0,

```
float pV0;  
ReadVelocity0(&pV0); //pV0 contains the motor 0 velocity in mm/sec
```

If you want set the desired position of motor0 to be 100mm from the current position, use the following code.

```
float pV0;  
pV0 = 100.0;  
SetPosition0(&pV0);
```

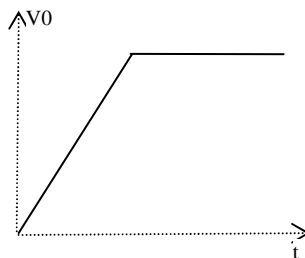
To read back the current position of the motor0,

```
float pV0;  
ReadPosition0(&pV0); //pV0 contains the motor 0 velocity in mm/sec
```

For applications without using encoder feedback, the velocity mode and position mode can not be used. However, the motor is still controllable by changing the PWM duty cycle to it. The follow code change the duty cycle of motor 0 to be 30%, and motor 1 to be -60%.

```
float pV0,pV1;  
pV0 = 0.3;  
pV1 = -0.6;  
SetPWM(&pV0, &pV1);
```

The iMicro motion controller applies acceleration limit on the velocity profile. When the user sets the desired velocity, the motor will reach the velocity at the maximum acceleration allowed.



If the maximum acceleration of motor 0 is to be set at 2000mm/sec/sec, use the following code,

```
float pV0;  
pV0 = 2000;  
SetAcceleration0(&pV0);
```