

io control

There are pins on the M5Stack-basic at the bottom of the robot arm and on the Atom at the end. The adsorption pump and other tools can be controlled by setting high and low levels of the pins through io control (For the numbers of the pins, see the pin labels attached at M5Stack-basic and Atom pins. The pins are common for input and output).

1 M5Stack-basic io control (m5)

1.1 Setting the high and low levels of output io

SetBasicOut(byte pin_number, byte pin_signal)

Return value: none

Parameter description: Parameter 1: pin number (basic output pin number); Parameter 2: state (0 - low level; 1- high level)

Case:

```
mc.SetBasicOut(2, 1);  
Thread.Sleep(100);  
mc.SetBasicOut(5, 1);  
Thread.Sleep(100);
```

1.2 Getting the state of input io

GetBasicIn(byte pin_number)

Return value: pin state (0-low level; 1-high level)

Parameter description: pin number (basic input pin number)

Case: set the output pin 2 to high level

```
Console.WriteLine(mc.GetBasicIn(35));  
Thread.Sleep(100);  
Console.WriteLine(mc.GetBasicIn(36));  
Thread.Sleep(100);
```

2 Atom io Control

Note: 320m5 has no atom io, so the module API is not used

2.1 Setitng the high and low levels of output io

SetDigitalOut(byte pin_number, byte pin_signal)

Return value: none

Parameter description: Parameter 1: pin number (atom output pin number); Parameter 2: state (0 - low level; 1- high level)

Case:

```
mc.SetDigitalOut(23, 0);
Thread.Sleep(100);
mc.SetDigitalOut(33, 0);
Thread.Sleep(100);
```

2.2 Getting the state of input io

GetDigitalIn(byte pin_number)

Return value: pin state (0-low level; 1-high level)

Parameter description: pin number (atom input pin number)

Case:

```
Console.WriteLine(mc.GetDigitalIn(19));
Thread.Sleep(100);
Console.WriteLine(mc.GetDigitalIn(22));
Thread.Sleep(100);
```

3 Complete use cases

```
using System;
using System.Threading;

namespace Mycobot.csharp
{
    class Test
    {
        static void Main(string[] args)
        {
            MyCobot mc = new MyCobot("COM57");
            mc.Open();
            Thread.Sleep(5000);

            //set basic output io
            /*mc.SetBasicOut(2, 1);
            Thread.Sleep(100);
            mc.SetBasicOut(5, 1);
            Thread.Sleep(100);
            mc.SetBasicOut(26, 1);
            Thread.Sleep(100);*/

            //get basic input io
            Console.WriteLine(mc.GetBasicIn(35));
            Thread.Sleep(100);
            Console.WriteLine(mc.GetBasicIn(36));
            Thread.Sleep(100);

            //set atom output io
            /*mc.SetDigitalOut(23, 0);
            Thread.Sleep(100);
            mc.SetDigitalOut(33, 0);
            Thread.Sleep(100);*/

            //get m5 input io
```

```
        /*Console.WriteLine(mc.GetDigitalIn(19));  
        Thread.Sleep(100);  
        Console.WriteLine(mc.GetDigitalIn(22));  
        Thread.Sleep(100);*/  
  
        mc.Close();  
    }  
}  
}
```