# Use cases

In this case, first get the current angles of all joints, then make the joint 1 returns to zero point, and finally get the values of two input pins on M5Stack-basic. The program.cs in the project is a complete use case program, which can be modified as needed on this basis:

```
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);
            // int[] angles = new[] {100, 100, 100, 100, 100, 100};
            // mc.SendAngles(angles, 50);
            var recv = mc.GetAngles();
             foreach (var v in recv)
             {
                Console.WriteLine(v);
             }

            // int[] coords = new[] {160, 160, 160, 0, 0, 0};
            // mc.SendCoords(coords, 90, 1);
            // Thread.Sleep(5000);
            // var recv = mc.GetCoords();
            // foreach (var v in recv)
            // {
            // Console.WriteLine(v);
            // }

            mc.SendOneAngle(1,0, 70);
            Thread.Sleep(100);
            /*var angle = new int[6];
            angle = mc.GetAngles();
            foreach (var v in angle)
                Console.WriteLine(v);
            // byte[] setColor = {0xfe, 0xfe, 0x05, 0x6a, 0xff, 0x00, 0x00,
0xfa};*/

            //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);*/

            //set gripper open or close 0--close 100-open max 0-100
            /*mc.setGripperValue(0, 10);
            Thread.Sleep(3000);
            mc.setGripperValue(50, 100);
            Thread.Sleep(3000);*/

            //get gripper state 0--close 1--open
            /*Console.WriteLine(mc.getGripperValue());*/
            mc.Close();
        }
    }
}
```