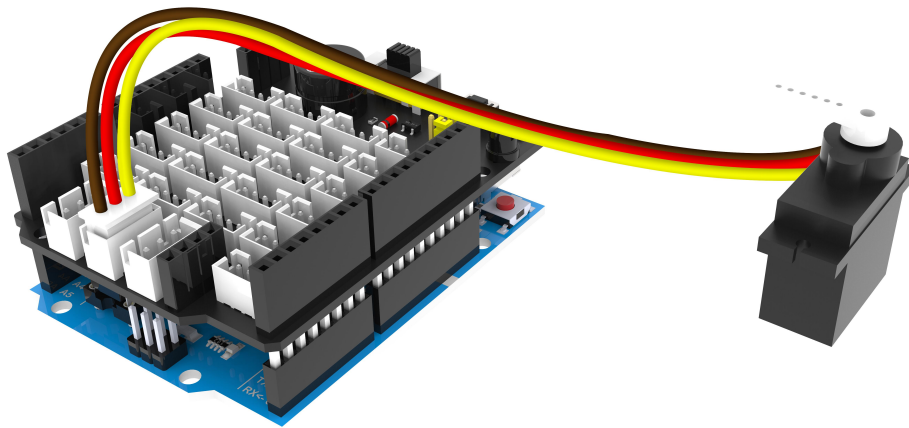


Experimental content: Send data via serial port to adjust the rotation angle of the servo

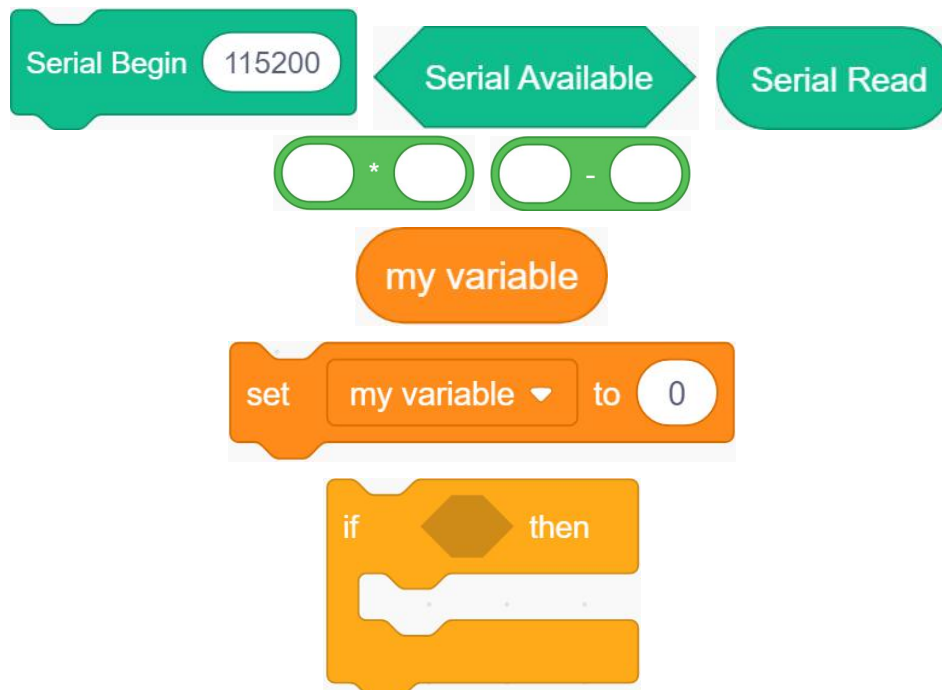
Experiment preparation: UNO board *1, Plugkit sensor expansion board *1, USB data cable *1, 9G Metal digital servo *1

Experimental wiring:



Experimental steps:

1. Adjust the rotation angle of the servo through the serial port, we need to use the following blocks in [arduino] and [calculation], [variable], and [control]



2. The main purpose of this lesson is to adjust the rotation angle of the servo by inputting characters through the serial port.

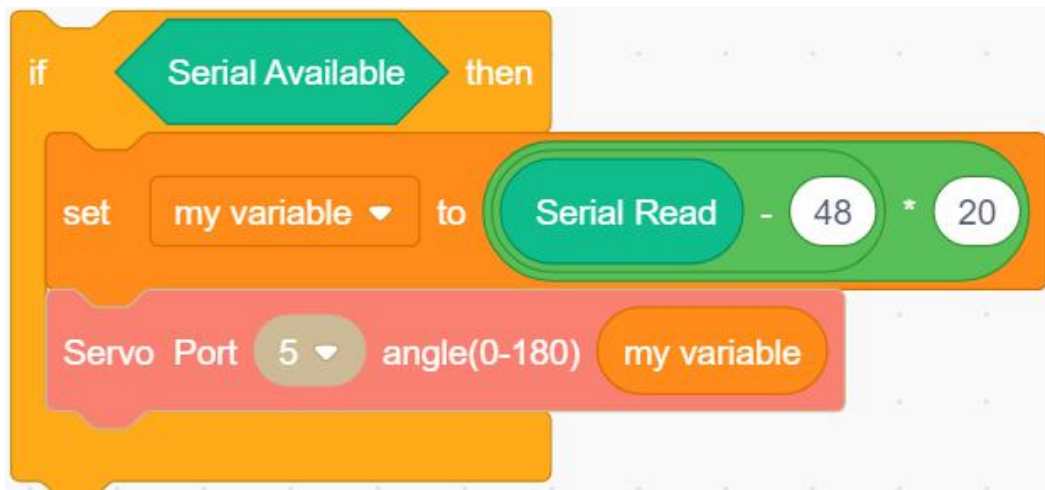
The computer directly sends characters through the serial port, such as sending 0. In fact, the character "0" is not decimal 0. The serial port actually reads the ASCII code value. Therefore, we need to convert the ASCII code value to a decimal number through operation. First simpler method to control the rotation of the servo by the result of the operation.

3. We need to use the characters "0" to "9". From the ASCII character code table, we can see that the characters "0" are converted into ASCII values 48, "1" is converted into ASCII values 49, and "2" is converted into ASCII values 50. ... We take the character "0" as the starting point 0, so we need to convert the ASCII value of the read character to a decimal number by subtracting 48. We can get the numbers 0-9, and then $0-9 \times 20$. So we can get nine rotation angles by enter the characters "0" ~ "9".

character	Rotation angle
"0"	0
"1"	20
"2"	40
"3"	60
...	...
"9"	180

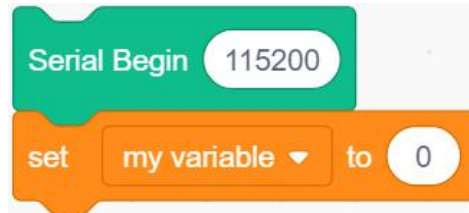


4. If there is data in the serial port, then execute to set my variable to (Serial Read data-48) * 20, modify the port of the servo to 5, and the rotation angle is my variable.

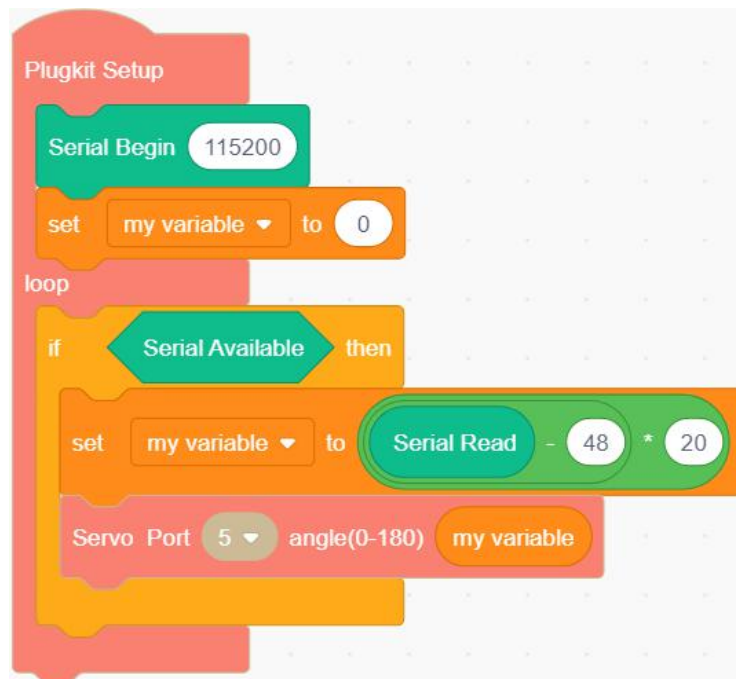


5. In addition, we need to initialize the baud rate of the serial port to 115200, set my variables to 0, and add them together.

Note: If you use [Plugkit] related serial port building blocks, you don't need to set the baud rate of the serial port. If you use [arduino], you need to set it.

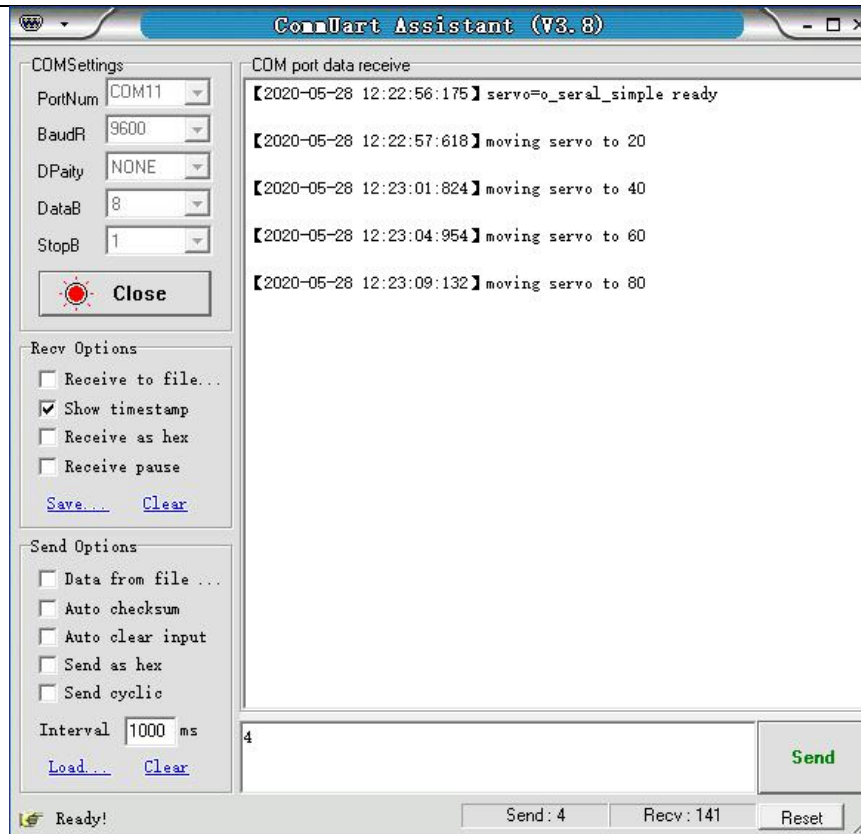


6. Put the block combination in step 4 into the loop, and put the block combination in step 5 into the setup initialization.



7. Compiling and uploading programs.

Experimental phenomena: Open the serial debugging assistant, set the baud rate to 115200, open the serial port. Then, send the numbers 0-9 in the sending area of the serial debugging assistant. Because the numbers sent by the serial debugging assistant are actually characters instead of decimal numbers, but after our conversion, it is equivalent to the decimal number is 0-9. When the number is 0, the servo angle is 0°, when the number is 1, the servo angle is 20°, when the number is 2, the servo angle is 40°, and so on.



Expand: There is another method to control the rotation angle of the servo through the serial port. It can rotate at any angle from 0 to 180. The specific method is as follows. It can be sent in hexadecimal, then, we can convert the specific angle (decimal) to hexadecimal and send it. Some angles are listed below.

Rotation angle (decimal)	Hex (Send)
0	0x00
1	0x01
2	0x02
11	0x0B
20	0x14
30	0x1E
90	0x5A
180	0xB4

Decimal to hexadecimal calculation method, that is, the integer is divided by 16 to take the remainder, until the quotient is 0, and then read from the last remainder to the first, such as 180: $180/16 = 11$ remainder 4, then use $11 / 16 = 0$ remainder 11, so the hexadecimal of decimal 180 is 0xB4. Read from the last remainder to the first remainder. The hexadecimal of decimal 11 is B.

Experimental phenomena: Open the serial port debugging assistant, set the baud rate to 115200. Then, open the serial port and set to send in hexadecimal in the serial port sending area. Next, if we send the hex number in the serial debugging assistant sending area, we can control servo.