**Me Infrared Receiver Decode**





**Overview**

Me IR Receiver includes an infrared signal receiver to receive the infrared signal transmitted from the distance. As the most widely used communication and remote control method at present, the infrared remote controller has the advantages of compact size, low power consumption, and powerful function. It is used with a variety of household appliances, audio equipment, air conditioning, robot motion control, car control and other intelligent control. In the environment of high pressure, radiation, poisonous gas, and dust, the infrared remote controller can effectively isolate the electrical interference. Its blue ID means that it has a double-digital signal port and needs to be connected to the port with blue ID on Makeblock Orion.

**Technical specifications**

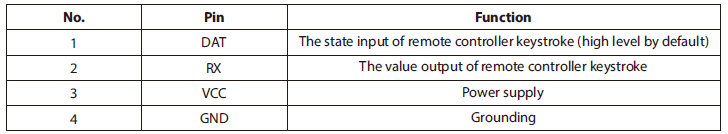
* Operating voltage: 4.8V~5.3V DC
* Operating current: 1.7~2.7 mA
* Receiving frequency: 38 KHz
* Peak wavelength: 980 nm
* Maximum receiving distance: 10 m
* Control mode: Double-digital port control
* Default baud rate: 9600
* Module size: 51 x 24 x 24.8mm (L x W x H)

**Functional characteristics**

* White area of module is the reference area to contact metal beams
* Makeblock remote controller can be used
* The remote controller should be targeted to infrared head when used in close range
* Provide an indicator for receiving signal
* Output from serial port when **NEC** IR protocol is used
* Anti-reverse protection – connecting the power supply inversely will not damage IC
* Support mBlock GUI programming, and applicable to users of all ages
* Adopt RJ25 port for easy connection
* Provide pin-type port to support most development boards including Arduino series

**Pin definition**

The port of Me IR Receiver has four pins, and their functions are as follows:

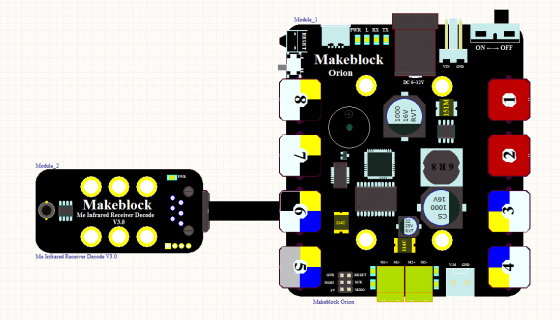


**Wiring mode**

● **Connecting with RJ25**

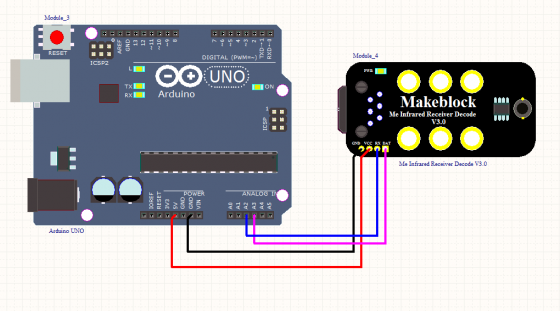
Since the port of Me IR Receiver has blue ID, you need to connect the port with blue ID on Makeblock Orion when using RJ25 port.

Taking Makeblock Orion as example, you can connect to ports No. 3, 4, 5, and 6 as follows:



● **Connecting with Dupont wire**

When the Dupont wire is used to connect the module to the Arduino UNO Baseboard, its RX and DAT pins should be connected to digital port as follows:

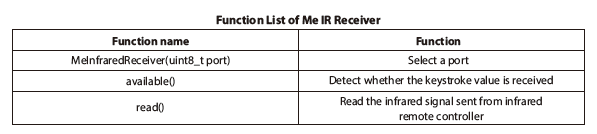


**Guide to programming**

● **Arduino programming**

If you use Arduino to write a program, the library Makeblock-Library-master should be invoked to control the Me IR Receiver. This is a routine to see the pressed keystroke of Me IR Remote Controller on the serial monitor through Arduino programming.





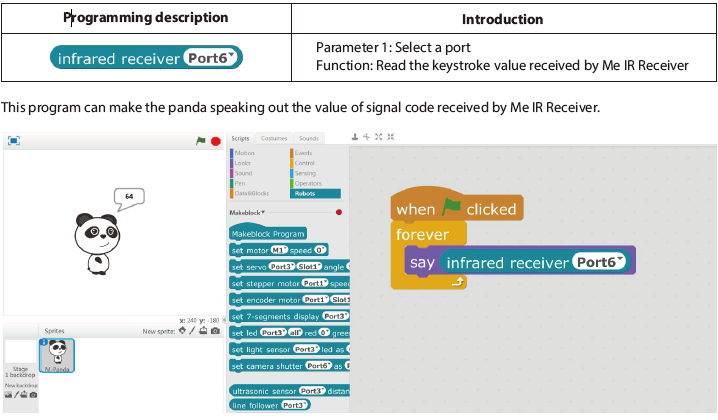
The function of the code segment is: to read the result of keystroke detection from Me IR Receiver and output the result to the serial monitor in Arduino IDE. Upload the code segment to the Makeblock Orion and click on the Arduino serial monitor, and you will see the running result as follows:



We can see that when the infrared signal is received from the infrared control, the Me IR Receiver reads the infrared signal for decoding, and then outputs to the serial port for displaying.

● **mBlock programming**

Me IR Receiver supports the mBlock programming environment and its instructions are introduced as follows:



**Principle analysis**

Infrared communication adopts infrared technology to implement close range secure communication and information transmission between two points, and it generally comprises the infrared transmitting system and infrared receiving system. Me IR Receiver includes an infrared integrated receiver head which contains a high-frequency filter circuit used to filter out carrier signal from the infrared synthetic signal, and then the signal is decoded into the module. When the infrared synthetic signal goes into the infrared receiver head, the digital code sent from infrared transmitter can be obtained at its output port (when the module receives effective infrared-coding data, STA becomes low level. If the key of infrared remote controller is pressed down continuously, STA will remain low level, and send data code repeatedly at the same time.

**Schematic**

