

K210 and microbit communication

K210 and microbit communication

- 1.Communication protocol description
 - 1.protocol definition
2. K210 and microbit communication
 - 2.1 Experimental premises
 - 2.2 Experimental wiring
 - 2.3 Main code analysis
 - 2.4 experimental phenomena
- 3.appendix
 - 3.1 K210 offline operation method
 - 3.2 Serial Port Assistant Data Analysis

1.Communication protocol description

1.protocol definition

	ed is optional for transmission	receives data with a length of 20 strings							
perimental routi	start	outine numb	data 1	Separator	...	data N	Separator	end	notes
ication protocol	\$	XX	XX	,	...	XX	,	#	Communication protocol format

analysis:

protocol definition	analysis
\$	start
Routine number	Two bytes, corresponding to routine ID number, with zeros added before values less than 10
Data volume	Number of data below
data	Data, separated by commas (,) after data, has as many commas as there are
#	end

2. K210 and microbit communication

2.1 Experimental premises

This tutorial uses microbit V2.0, and K210 requires running the program in **K210-AI (microbit)** to start the experiment

microbit *1

Eyu clip *3

Several DuPont Lines

K210 perspective module * 1 (requires SD card (with AI model inside) and camera)

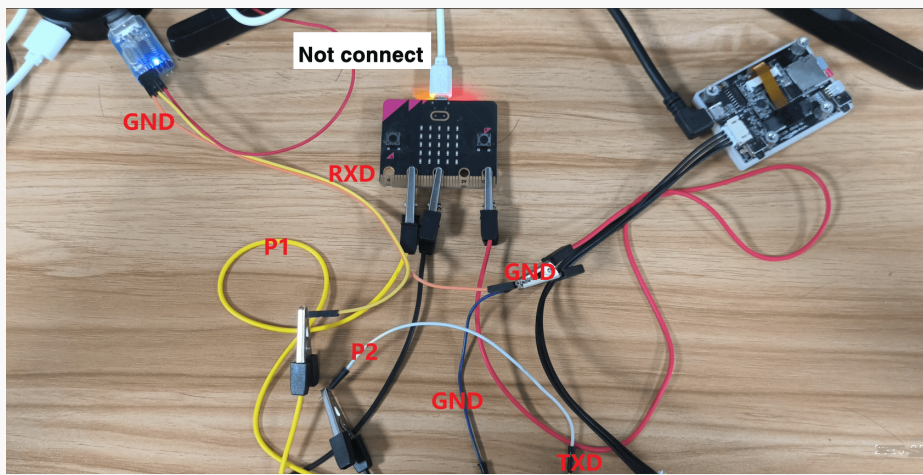
USB to TTL module*1

2.2 Experimental wiring

microbit	USB to TTL module
P1	RXD
GND	GND

microbit	K210 perspective module
P2	TXD
GND	GND

Wiring as shown in the diagram:



2.3 Main code analysis

```

on start
  let strr = ""
  serial
  redirect to
  TX P1
  RX P2
  at baud rate 115200

forever
  if strr != "" then
    show string strr
    set strr to ""
  clear screen

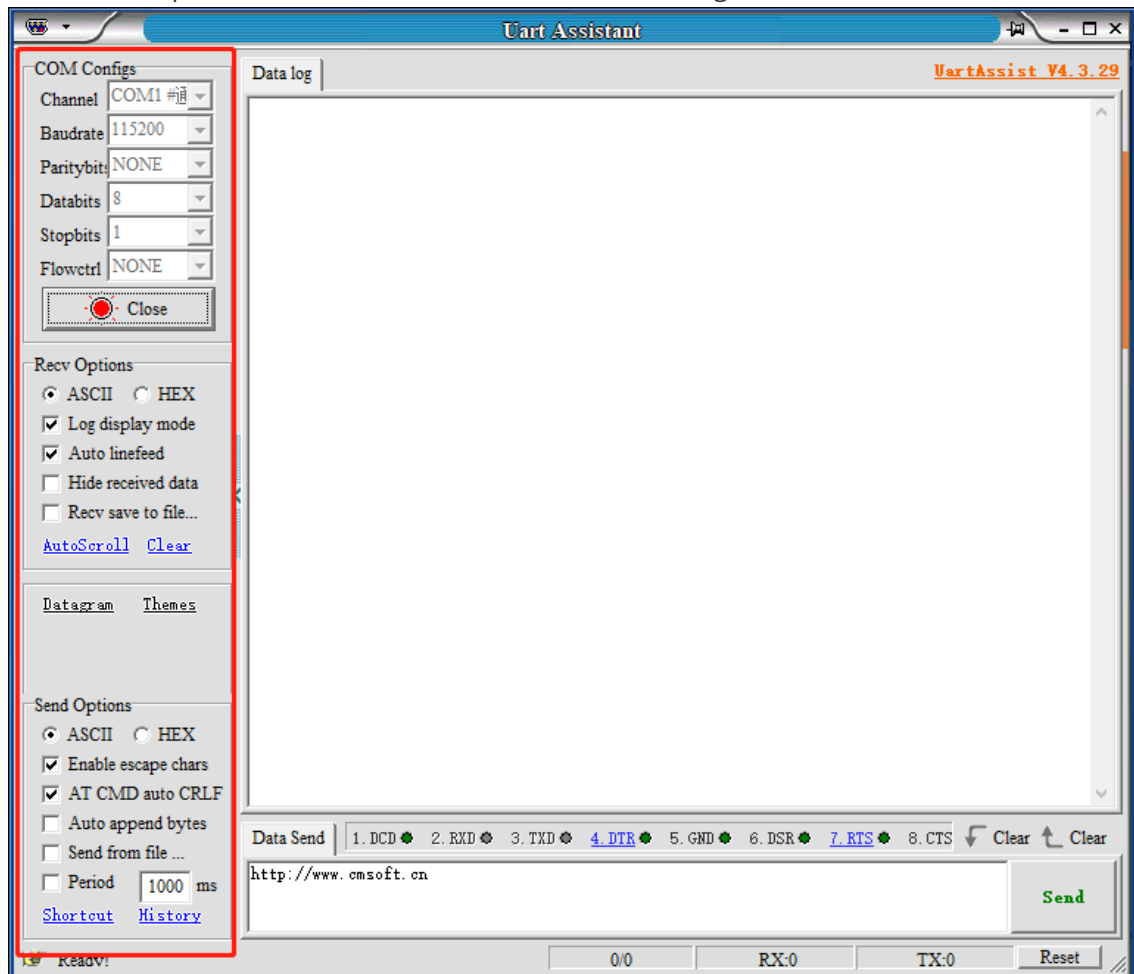
function deal_data lenn
  let k: number;
  set class_num to join j get value at 1 j get value at 2 - +
  set index_flag to 1
  if class_num == "01" or class_num == "05" or class_num == "06" then
    set k to 3
    while k < lenn
      do
        if j get value at k == "," then
          if index_flag == 1 then
            set x to join x j get value at k - +
          if index_flag == 2 then
            set y to join y j get value at k - +
          if index_flag == 3 then
            set w to join w j get value at k - +
        else
          set index_flag to index_flag + 1
    serial on data received $
      set j to serial read until #
      set lenn2 to length of j - 1
      call deal_data lenn2
  
```

From the source code, it can be seen that:

1. When serial communication receives information with \$ as the header and # as the tail, it will call deal_ The data block is used for data analysis.
2. Deal_ The parsing of data is based on the protocol, obtaining key information and displaying it on the screen and printing serial port.
3. For those interested in the source code, you can <https://makecode.microbit.org> Open the source code provided in this tutorial on the website for learning.

2.4 experimental phenomena

1. After connecting the cable, the K210 perspective module can be run offline or connected to the camv ide, but some experiments need to be run offline. Please refer to the appendix for the methods of offline operation
2. Set the serial port assistant to the interface shown in the figure



3. Download the hex program of this tutorial into microbit
4. After starting recognition, you can see that the serial assistant prints something and the microbit screen also scrolls to display the received information. The experimental results in the following figure show the phenomenon of object detection

[illegible]

The serial assistant prints out the recognized object labels

3.appendix

3.1 K210 offline operation method

- First, put the model needed for recognition on the SD card
[Import model to SD card](#)
- Save the required AI program to the SD card and rename it main.py
- Power on K210 to run the main.py program on the SD card

3.2 Serial Port Assistant Data Analysis

x: Abscissa

y: Ordinate

w: Width

h: Length

ID: Identified object label

Str: Identified content information