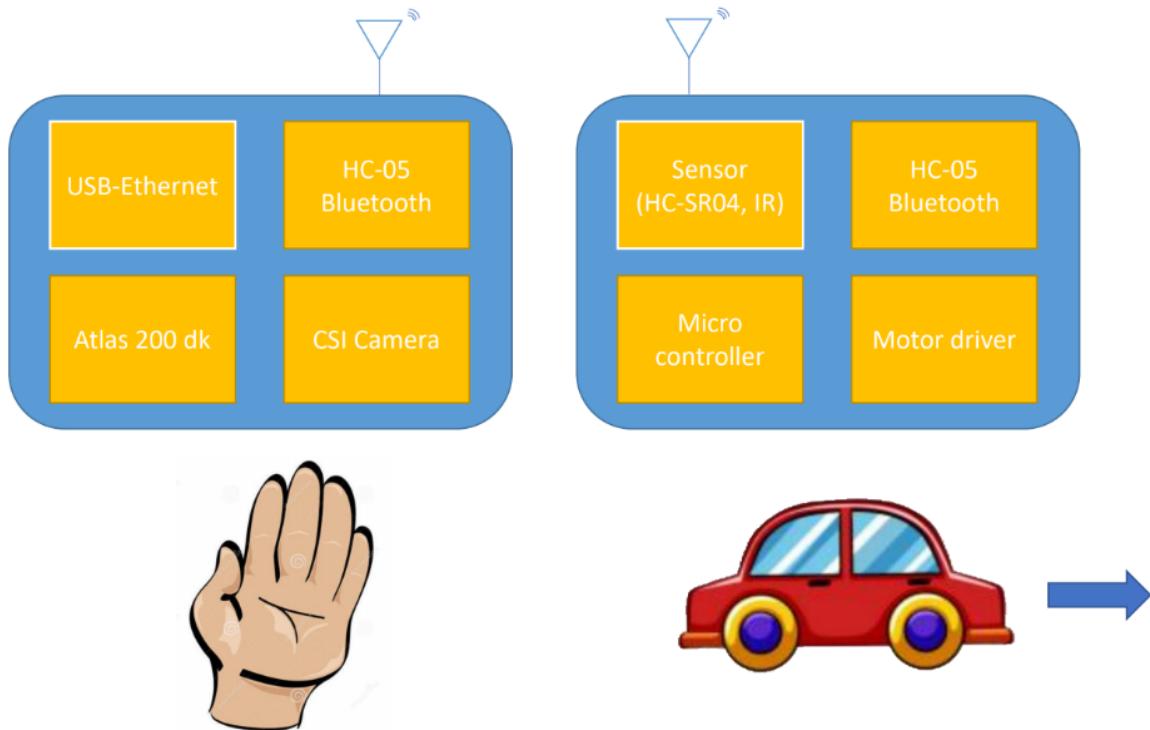


Lesson-1 Built your smart robot car with Bluetooth modules

Introduction

The following figure shows the block diagram of the hand pose remote-control car based on Atlas 200dk. The system is mainly composed of two parts: hand pose recognition (left) and wireless remote-control car (right). First, the camera will capture the picture of the palm, after Atlas 200dk processing, identify and send the corresponding instructions to the Remote-control car through the HC-08 Bluetooth module on 200dk. The car controls the forward or reverse rotation of the motor according to the corresponding instructions.



Atlas 200dk Setup

The following figure shows the basic components before development of Atlas 200dk including the developing board, power adaptor, USC Type-C cable, SD card, CSI camera and its holder.



You can set up your Atlas 200dk by the following steps:

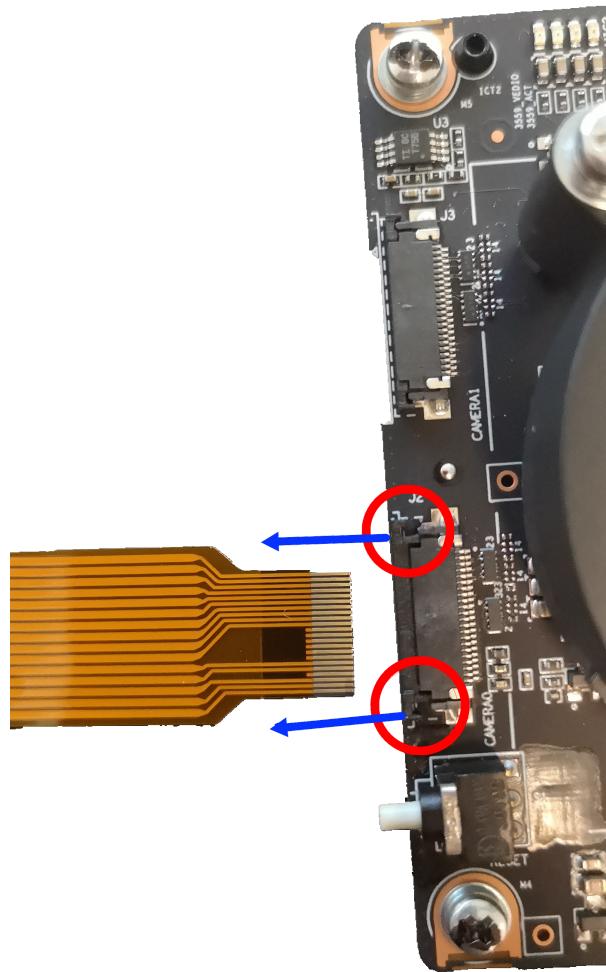
1. Write the image file to the SD card with [Win32 Disk Imager](#);
2. Replace the 15 pins CSI camera ribbon flexible cable with new FPC cable, as is shown below;



3. Assemble the CSI Camera with Camera Holder;
4. Remove the cover of the board;



5. Loosen the CAMERA 0 Connector and connect the camera to the board;



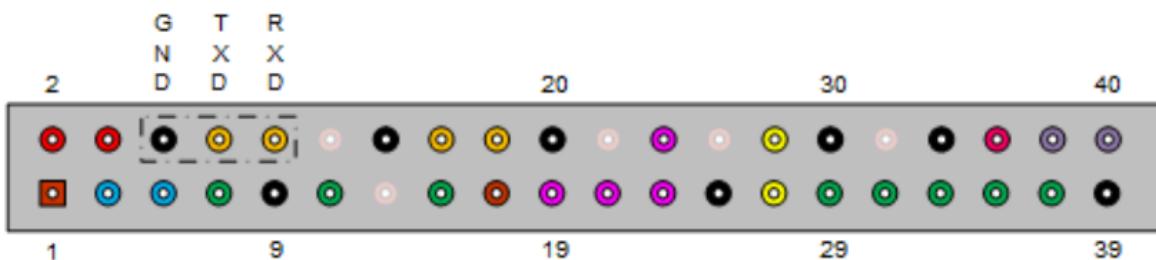
Note: The figure shows the camera connector of IT21DMDA, while the IT21VDMB uses another 50 pin MIPI-CSI connector.

6. Connect the Bluetooth module to the board UART1 (*UART0 used for Linux Shell*):

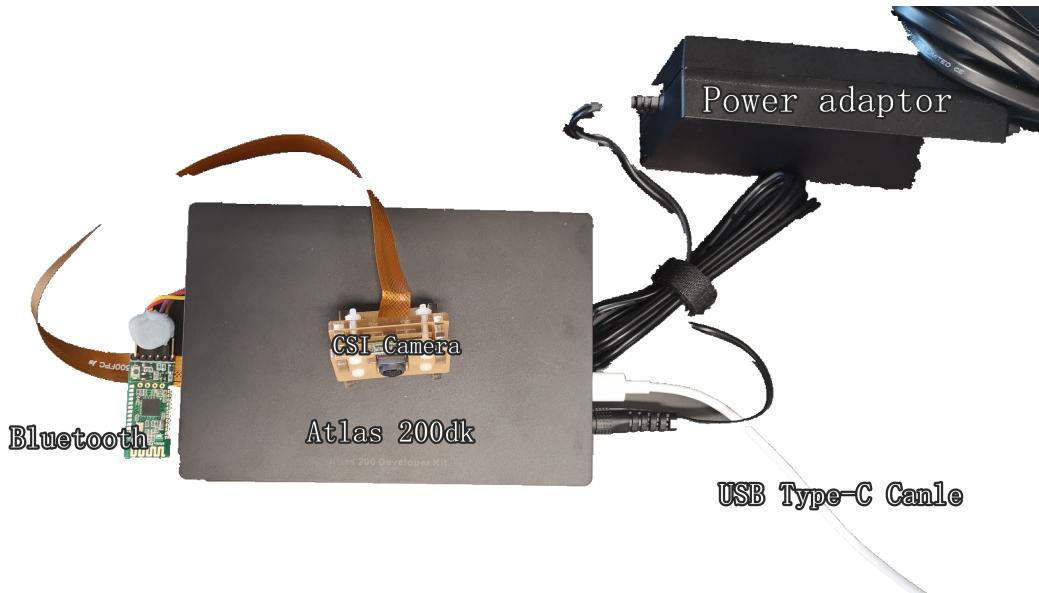
	Bluetooth Module Pin	Atlas 200 dk Pin
1.	VCC	+5.0V
2.	GND	GND
3.	TXD	RXD1
4.	RXD	TXD1

Atlas 200 Pin definition

Pin	Name	Voltage	Pin	Name	Voltage
1	+3.3V	3.3V	2	+5.0V	5V
3	I2C-SDA	3.3V	4	+5.0V	5V
5	I2C-SCL	3.3V	6	GND	-
7	GPIO0	3.3V	8	TXD0	3.3V
9	GND	-	10	RXD0	3.3V
11	GPIO1	3.3V	12	NC	-
13	NC	-	14	GND	-
15	GPIO2	3.3V	16	TXD1	3.3V
17	+3.3V	3.3V	18	RXD1	3.3V
19	SPI-MOSI	3.3V	20	GND	-
21	SPI-MISO	3.3V	22	NC	-
23	SPI-CLK	3.3V	24	SPI-CS	3.3V
25	GND	-	26	GPIO10	3.3V
27	GPIO8	3.3V	28	GPIO9	3.3V
29	GPIO3	3.3V	30	GND	-
31	GPIO4	3.3V	32	NC	-
33	GPIO5	3.3V	34	GND	-
35	GPIO6	3.3V	36	+1.8V	1.8V
37	GPIO7	3.3V	38	TXD-3559	3.3V
39	GND	-	40	RXD-3559	3.3V



7. Finish Atlas hardware setup and setup software by the link(<https://github.com/Atlas200dk/sample-handposeRCcar/tree/c73/sample-handposeRC/Atlas200DK/sample-handposeRC>);



Remote-control car setup

- The basic framework of Remote-control car assembling, please refer to [User Manual\(EN\)20200116.pdf](#). (Don't connect the Bluetooth module to the socket connector);
- Upload the program to the microcontroller, program link (<https://github.com/Atlas200dk/sample-handposeRCcar/tree/c73/sample-handposeRC/Arduino>);
- Then, connect the Bluetooth module to the Remote-control car (*You cannot upload the program, when the Bluetooth module connected*);

	Bluetooth Module Pin	Remote-control car
1.	VCC	+5.0V
2.	GND	GND
3.	TXD	RXD
4.	RXD	TXD

Finish the remote-control car setup

Bluetooth Setup HC08(optional)

Bluetooth introduction

HC-08 embedded Bluetooth serial communication module (*can be short for module*) has two work modes: **Order-response work mode** and **Automatic connection work mode**.

- Automatic connection work mode:** it will follow the default way set lastly to transmit the data automatically (Baud rate: 9600). Once the module connect to other Bluetooth devices (e.g. Mobile phone, another module), it will automatically be configured to this mode (**LED lights always**).
- Order-response work mode:** User can send the AT command to the module to set the control parameters and sent control order (Baud rate: 9600). Before the Bluetooth module connection, it works in Order-response work mode. (**LED blink**)

How to pair two Bluetooth module

There are 2 roles, master and slaver, in the Bluetooth communication. Therefore the modules should be configured as the master and slaver separately. Meanwhile, both of module should be configured with the same **Search UUID, Service UUID, and Characteristic UUID**. Then when both of modules power on, they will automatically connect.

How to configure as the slaver

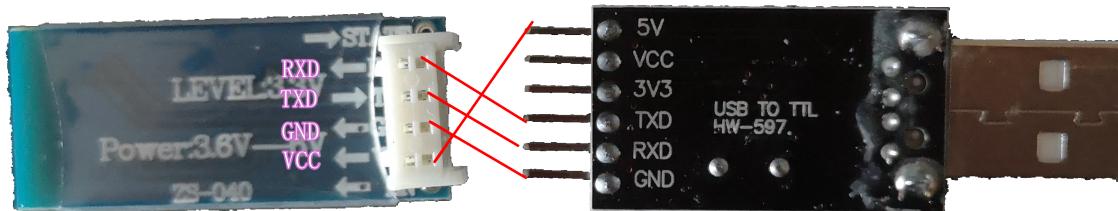
1. Check the module work in *Order-response work mode* by `AT` ;
2. Clear Address by `AT+CLEAR` (*HC08 only supports one master to one slaver, before re-configuration the previous address must be clear*)
3. Set the module as slave;
4. Set 3 UUID separately;
5. Set the module name;

How to configure as the master

1. Check the module work in *Order-response work mode* by `AT` ;
2. Clear Address by `AT+CLEAR` (*HC08 only supports one master to one slaver, before re-configuration the previous address must be clear*)
3. Set the module as master;
4. Set 3 UUID separately;
5. Set the module name;

How to use the demo [Bluetooth Setup HC08.py](#)

1. Prepare 2 USB to UART TTL module and 2 HC08, and connect as following:



	Bluetooth Module Pin	USB to UART (TTL) module
1.	VCC	+5.0V
2.	GND	GND
3.	TXD	RXD
4.	RXD	TXD

- The alternative solution is to use the *USB to UART* function in *Arduino*
- You can also configure 2 modules one by one with only one *USB-UART* module.

2. Set `checkUART = 1` and *Run*, you will get

```
COM11 - USB Serial Port (COM11)
COM5 - USB-SERIAL CH340 (COM5)
```

3. Set as following and Run:

```
CheckUART = 0
Configure = 1
# Port_M,Port_S depends on the result in Step 1;
Port_M = "COM5"
Port_S = "COM11"
BLE_Name = "XXXX"
LUUID = "123F" #4-digital hexadecimal number
SUUID = "123F" #4-digital hexadecimal number
TUUID = "123F" #4-digital hexadecimal number
#It is not necessary for LUUID,SUUID,TUUID to be configure with the same number;
#You'd better to select a random 4-digital hexadecimal number as
LUUID,SUUID,TUUID;
```

4. Re-connect 2 Bluetooth modules. If they are connected, the LED will always turn on;

5. Set Configure = 0 and Run, the program will automatically test the connection between the modules;