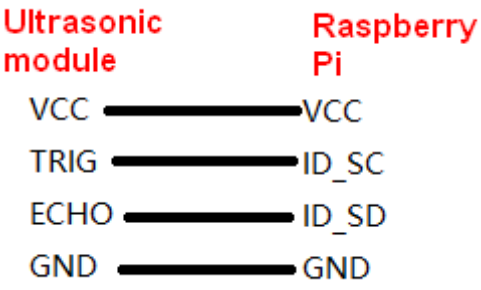
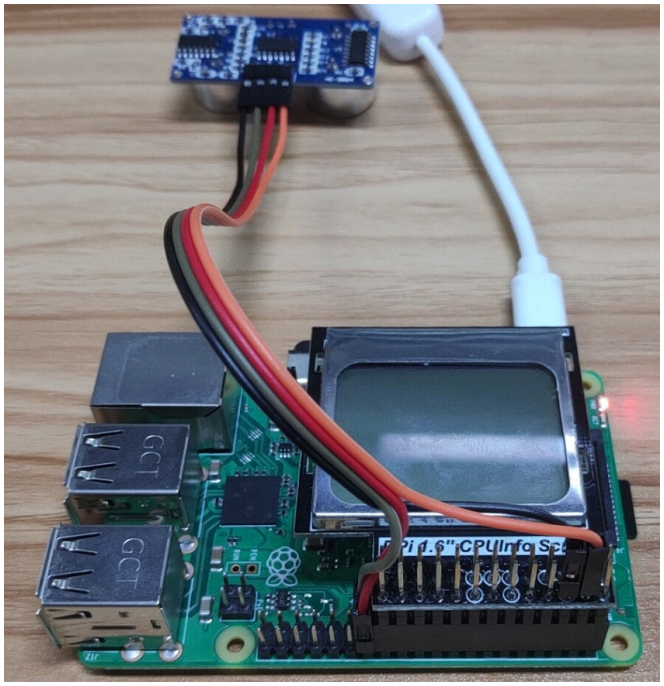
7.Display ultrasonic distance data

After CPU Info LCD screen is correctly inserted into the Raspberry Pi, you need to compile and run the program to display it normally. This experiment is used to display ultrasonic distance data.

1.Ultrasonic wiring





2. Install the wringPi library

CPU Info LCD screen is used for data communication through the GPIO port of the Raspberry Pi, so we must install the wiringPi library file.

Enter the following command to install the wringPi library. Users who have already installed the wiringPi library can ignore this step.

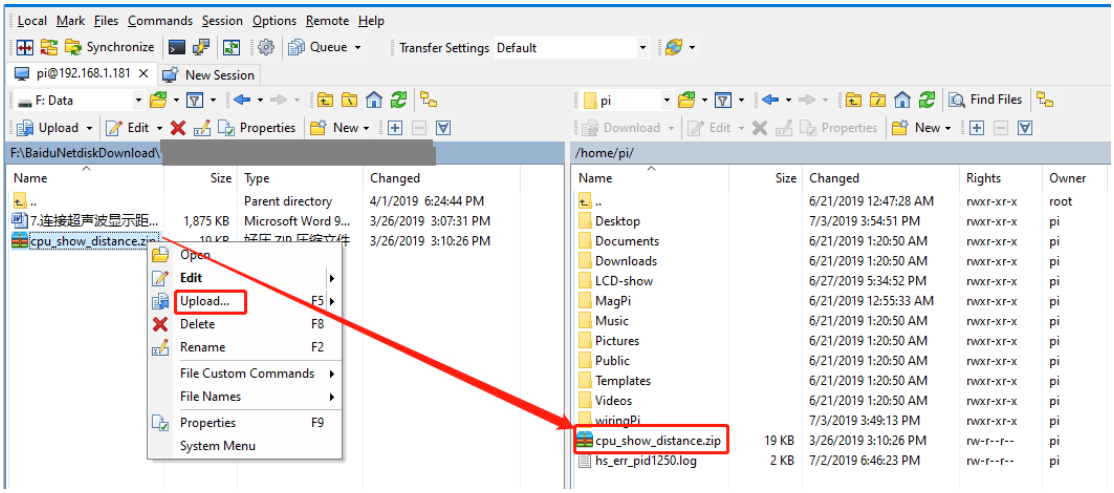
```
cd ~
git clone git://git.drogon.net/wiringPi
cd wiringPi
./build
```

3.Install Drive

1) Transfer the driver file to the Raspberry Pi

You need to install the Winscp tool on your computer. After connecting to the Raspberry Pi, transfer the **cpu_show_distance.zip** package from this folder to the pi directory of the Raspberry Pi.

As shown blew, drag and drop **cpu_show_distance.zip** directly into the Raspberry Pi system.



2)Extract file

Open the Raspberry Pi terminal and find the **cpu_show_distance.zip** file.

Enter command:

ls

```
pi@raspberrypi:~ $ ls
cpu_show_distance.zip  Downloads  matchbox-keyboard  Public        voice-engine
Desktop                env        Music              python_games
Documents              MagPi      Pictures           Templates
pi@raspberrypi:~ $
```

Enter command:

unzip cpu_show_distance.zip

```
pi@raspberrypi:~ $ unzip cpu_show_distance.zip
Archive:  cpu_show_distance.zip
   creating: cpu_show_distance/
  inflating: cpu_show_distance/PCD8544.c
  inflating: cpu_show_distance/README.txt
  inflating: cpu_show_distance/PCD8544.h
  inflating: cpu_show_distance/cpushow_distance
  inflating: cpu_show_distance/pcd8544_rpi.c
pi@raspberrypi:~ $
```

3) Enter the program folder

cd ~/cpu_show_distance

ls

```
pi@raspberrypi:~/cpu_show_distance $ ls
cpushow_distance  PCD8544.c  PCD8544.h  pcd8544_rpi.c  README.txt
pi@raspberrypi:~/cpu_show_distance $
```

4) Compiler file

Enter command:

cc -o cpushow_distance pcd8544_rpi.c PCD8544.c -L/usr/local/lib -lwiringPi

```
pi@raspberrypi:~/cpu_show_distance $ cc -o cpushow_distance pcd8544_rpi.c PCD854
4.c  -L/usr/local/lib -lwiringPi
pcd8544_rpi.c: In function 'Distance':
pcd8544_rpi.c:78:3: warning: implicit declaration of function 'gettimeofday' [-W
implicit-function-declaration]
   gettimeofday(&tv3, NULL);
   ^~~~~~~~~~~~
pi@raspberrypi:~/cpu_show_distance $
```

5) Running procedure

Enter command:

./cpushow_distance

```
pi@raspberrypi:~/cpu_show_distance $ ./cpushow_distance
Raspberry Pi PCD8544 sysinfo display
=====================================
distance= 10cm
distance= 15cm
distance= 6cm
distance= 8cm
distance= 10cm
distance= 8cm
distance= 7cm
distance= 9cm
distance= 11cm
distance= 12cm
distance= 15cm
```

The system will prompt “Raspberry Pi PCD8544 sysinfo display” and display the following on the CPU Info screen.

4.Code analysis

Enter command:

nano pcd8544_rpi.c

This command is to open pcd8544_rpi.c

1) The following sections are the LCD pin settings：

```
// pin setup
int _din = 1;
int _sclk = 0;
int _dc = 2;
int _rst = 4;
int _cs = 3;
```

The following sections are the ultrasonic pin settings:

```
//Define the EchoPin connect to wiringPi port 30 of Raspberry pi
//Define the TrigPin connect to wiringPi port 31 of Raspberry pi
int EchoPin = 30;
int TrigPin = 31;
```

2) Ultrasonic data acquisition

```
float Distance()
{
  float distance;
  struct timeval tv1;
  struct timeval tv2;
  struct timeval tv3;
  struct timeval tv4;
  long start, stop;

  digitalWrite(TrigPin, LOW);
  delayMicroseconds(2);
  //Input a high level of at least 10 US to the Trig pin
  digitalWrite(TrigPin, HIGH);
  delayMicroseconds(15);
  digitalWrite(TrigPin, LOW);

  //Add a timeout retest mechanism to prevent the program did not detect a level change,
  //get stuck in an infinite loop
  gettimeofday(&tv3, NULL);
  //Timeout retest mechanism is starting
  start = tv3.tv_sec * 1000000 + tv3.tv_usec;
  while(!digitalRead(EchoPin) == 1)
  {
    gettimeofday(&tv4, NULL);
    //Timeout retest mechanism is ending
    stop = tv4.tv_sec * 1000000 + tv4.tv_usec;
    //Maximum time value (5m): 10/340=0.03s
    if ((stop - start) > 30000)
    {
      return -1;
    }
  }

  gettimeofday(&tv1, NULL);
  start = tv1.tv_sec*1000000+tv1.tv_usec;
  while(!digitalRead(EchoPin) == 0)
  {
    gettimeofday(&tv3,NULL);
    stop = tv3.tv_sec*1000000+tv3.tv_usec;
    if ((stop - start) > 30000)
    {
      return -1;
    }
  }
  gettimeofday(&tv2, NULL);

  start = tv1.tv_sec * 1000000 + tv1.tv_usec;
  stop = tv2.tv_sec * 1000000 + tv2.tv_usec;

  distance = (float)(stop - start)/1000000 * 34000 / 2;
  return distance;
}
```

3) Bubble sorting

```
//Bubble sorting
void bubble(unsigned long *a, int n)
{
  int i, j, temp;
  for (i = 0; i < n - 1; i++)
  {
    for (j = i + 1; j < n; j++)
    {
      if (a[i] > a[j])
      {
        temp = a[i];
        a[i] = a[j];
        a[j] = temp;
      }
    }
  }
}

//Remove the maximum, minimum of the 5 datas
//and get average values of 3 datas to improve accuracy of test
float Distance_test()
{
  float distance;
  unsigned long ultrasonic[5] = {0};
  int num = 0;
  while (num < 5)
  {
    distance = Distance();

    while((int)distance == -1)
    {
      distance = Distance();
    }
    //Filter out data greater than 500 or
    while ( (int)distance >= 500 || (int)distance == 0)
    {
      distance = Distance();
    }
    ultrasonic[num] = distance;
    num++;
    delay(10);
  }
  num = 0;
  //Bubble sorting
  bubble(ultrasonic, 5);
  //Average
  distance = (ultrasonic[1] + ultrasonic[2] + ultrasonic[3]) / 3;

  printf("distance= %.0fcm\n",distance);
  return distance;
}
```

4）The main function is mainly divided into two parts. The front part is the initialization operation of the display screen and the ultrasonic module, and the latter part is a for loop to acquire and display data.

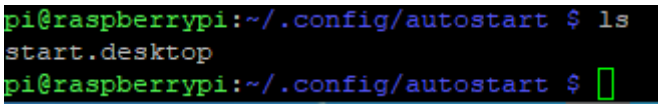
```
int main (void)
{
  // print infos
  printf("Raspberry Pi PCD8544 sysinfo display\n");
  printf("=======================================\n");

  // check wiringPi setup
  if (wiringPiSetup() == -1)
  {
    printf("wiringPi-Error\n");
    exit(1);
  }
  //Initialize ultrasonic pin
  pinMode(EchoPin, INPUT);
  pinMode(TrigPin, OUTPUT);

  //CurrentDistance
  char CurrentDistance[15];
  float distance = 0;

  // init and clear lcd
  LCDInit(_sclk, _din, _dc, _cs, _rst, contrast);
  LCDclear();
  delay(500);

  for (;;)
  {
    // clear lcd
    LCDclear();
    //get current distance data
    distance = Distance_test();

    sprintf(CurrentDistance,"distance:%.0fcm",distance);

    // build screen
    //LCDdrawstring(0, 0, "Raspberry Pi:");

    LCDdrawstring(0, 1, "Hello YahBoom!");
    LCDdrawline(0, 10, 83, 10, BLACK);

    LCDdrawstring(0, 21, CurrentDistance);

    LCDdisplay();

    delay(1000);
  }

  return 0;
}
```

Note: If you have added a boot-up user, first move the xx.desktop file displayed on the 1.6-inch screen in the /home/pi/.config/autostart folder to the pi directory.

If you do not close a program that has already been run, the screen will always change due to conflicts after the program runs.

For example, there is a file driver.desktop that drives a 1.6-inch screen in the

/home/pi/.config/autostart folder.



We need to move **start.desktop** to the pi directory:

Enter command:
mv /home/pi/.config/autostart/start.desktop /home/pi

Then we can enter command:

sudo reboot

This command is to restart the Raspberry Pi.