硬件代码。

硬件：超声波及报警装置。

// defines pins numbers

const int trigPin = 9;

const int echoPin = 10;

float cm;

// defines variables

long duration;

int distance;

void setup()

{

Serial.begin(9600);

pinMode(11, OUTPUT); //发红色光

pinMode(12, OUTPUT); //发蓝色光

pinMode(13, OUTPUT); //发绿色光

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(6, OUTPUT); //VIBRATOR

pinMode(3,OUTPUT); //BUZZER

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

}

void loop()

{

//发一个10ms的高脉冲去触发TrigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

cm = pulseIn(echoPin, HIGH) / 58.0; //算成厘米

cm = (int(cm \* 100.0)) / 100.0; //保留两位小数

Serial.print(cm);

Serial.print("cm");

Serial.println();

delay(1000);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration = pulseIn(echoPin, HIGH);

// Calculating the distance

distance= duration\*0.034/2;

if(cm<25)

{

Serial.print("warning");

digitalWrite(11,HIGH); digitalWrite(12,LOW); digitalWrite(13,LOW);

analogWrite(6,140);

tone(3,1000,1000);

}

else if(cm<50)

{

Serial.print("narrow");

digitalWrite(11,LOW); digitalWrite(12,HIGH); digitalWrite(13,LOW);

analogWrite(6,90);

tone(3,1000,1000);

}

else if(cm<75)

{

Serial.print("OK");

digitalWrite(11,LOW); digitalWrite(12,LOW); digitalWrite(13,HIGH);

analogWrite(6,60);

analogWrite(3,255);

}

else if(cm<100)

{

Serial.print("safe");

digitalWrite(11,HIGH); digitalWrite(12,LOW); digitalWrite(13,LOW);

analogWrite(6,30);

analogWrite(3,255);

}

else

{

Serial.print("very safe");

digitalWrite(11,LOW); digitalWrite(12,LOW); digitalWrite(13,HIGH);

analogWrite(6,0);

analogWrite(3,0);

noTone(3);

}

}

**GPS 的发送**

#include<Servo.h>

#include <TimerOne.h>

#define GPSget Serial2

#define DebugSerial Serial

#define GprsSerail Serial3

#define GPRMC\_TERM "$GPRMC," //定义要解析的指令，因为这条指令包含定位和时间信息

#define Success 1U

#define Failure 0U

int L=28;

unsigned long Time\_Cont = 0; //定时器计数器

char a\_y[]="1";

char a\_n[]="0";

const unsigned int gprsRxBufferLength = 600;

char gprsRxBuffer[gprsRxBufferLength];

unsigned int gprsBufferCount = 0;

char OneNetServer[] = "api.heclouds.com"; //不需要修改

char device\_id[] = "542005309"; //修改为自己的设备ID

char API\_KEY[] = "IvYhykyRqpoN6zhm5Wz1HKy=NJg="; //修改为自己的API\_KEY

char sensor\_gps[] = "loc";

char lon\_str[15];

char lat\_str[15];

char nmeaSentence[68];

String latitude; //纬度

String longitude; //经度

int isfull=0;

Servo myServo;

int buzzer=2;

int red\_line=6;

int TrigPin3=7;

int EchoPin3=8;

int ledPin=12;

int button=13;

int distance3;

int Switch;

void setup() {

pinMode(EchoPin3,INPUT);

pinMode(TrigPin3,OUTPUT);

pinMode(buzzer,OUTPUT);

pinMode(ledPin,OUTPUT);

pinMode(red\_line,INPUT);

pinMode(button,INPUT\_PULLUP);

myServo.attach(9);

myServo.write(90);

DebugSerial.begin(19200);

GprsSerail.begin(19200);

GPSget.begin(9600);

Timer1.initialize(1000);

Timer1.attachInterrupt(Timer1\_handler);

initGprs();

DebugSerial.println("\r\nsetup end!");

}

void loop() {

distance3=ping3();

if(distance3<=5&&distance3>2)

isfull=isfull+1;

int buttonState=digitalRead(button);

int red\_line\_get=digitalRead(red\_line);

if(buttonState==1)

{

myServo.write(21);/\*垃圾桶打开\*/

Switch=1;

isfull=-1300;

}

else

{

if((isfull>270)&&(Switch==0))

{

delay(1000);

//initGprs();

while(buttonState==0)

{

for(int k=0;k<10;k++)

alarm();

for(int pl=0;pl<2;pl++)

gpsToOnenet(1);

buttonState=digitalRead(button);

}

}

else

{

if(red\_line\_get==0)

{

myServo.write(21);

delay(3200);

isfull=0;

for(int i=21;i<=86;i++)

{

myServo.write(i);

delay(55);

}

gpsToOnenet(0);

}

else

{

if(Switch==1)

{

delay(1000);

for(int i=21;i<=86;i++)

{

myServo.write(i);

delay(55);

}

isfull=0;

Switch=0;

}

}

}

}

}

int ping3()

{

int duration3;

digitalWrite(TrigPin3,LOW);

delayMicroseconds(2);

digitalWrite(TrigPin3,HIGH);

delayMicroseconds(10);

digitalWrite(TrigPin3,LOW);

duration3=pulseIn(EchoPin3,HIGH)/58;

//Serial.println(duration3);

return duration3;

}

void alarm()

{

digitalWrite(buzzer,HIGH);

digitalWrite(ledPin,HIGH);

delay(100);

digitalWrite(buzzer,LOW);

digitalWrite(ledPin,LOW);

delay(100);

}

void gpsToOnenet(int is){

// For one second we parse GPS data and report some key values

for (unsigned long start = millis(); millis() - start < 1000;) //一秒钟内不停扫描GPS信息

{

while (GPSget.available()) //串口获取到数据开始解析

{

char c = GPSget.read(); //读取一个字节获取的数据

switch(c) //判断该字节的值

{

case '$': //若是$，则说明是一帧数据的开始

GPSget.readBytesUntil('\*', nmeaSentence, 67); //读取接下来的数据，存放在nmeaSentence字符数组中，最大存放67个字节

// GPSget.println(nmeaSentence);

latitude = parseGprmcLat(nmeaSentence); //获取纬度值 String

longitude = parseGprmcLon(nmeaSentence);//获取经度值 String

if((latitude > "")&&(longitude > ""))

{

for(int c=0;c<=9;c++)

lon\_str[c]=longitude.charAt(c);

for(int b=0;b<=8;b++)

lat\_str[b]=latitude.charAt(b);

DebugSerial.println(lat\_str);

DebugSerial.println(lon\_str);

postGpsDataToOneNet(API\_KEY,device\_id,sensor\_gps,lon\_str,lat\_str,is); //开始传输数据

}

}

}

}

delay(2000);

delay(5000);

}

double longitudeToOnenetFormat(char\* lon\_str\_temp)

{

double lon\_temp = 0;

long lon\_Onenet = 0;

int dd\_int = 0;

long mm\_int = 0;

double lon\_Onenet\_double = 0;

lon\_temp = atof(lon\_str\_temp);

lon\_Onenet =lon\_temp\*100000; //转换为整数

dd\_int = lon\_Onenet/10000000; //取出dd

mm\_int = lon\_Onenet%10000000; //取出MM部分

lon\_Onenet\_double = dd\_int + (double)mm\_int/60/100000;//换算为Onenet格式

return lon\_Onenet\_double;

}

double latitudeToOnenetFormat(char\* lat\_str\_temp)

{

double lat\_temp = 0;

long lat\_Onenet = 0;

int dd\_int = 0;

long mm\_int = 0;

double lat\_Onenet\_double = 0;

lat\_temp = atof(lat\_str\_temp);

lat\_Onenet =lat\_temp\*100000; //转换为整数

dd\_int = lat\_Onenet/10000000; //取出dd

mm\_int = lat\_Onenet%10000000; //取出MM部分

lat\_Onenet\_double = dd\_int + (double)mm\_int/60/100000;//换算为Onenet格式

return lat\_Onenet\_double;

}

void postGpsDataToOneNet(char\* API\_VALUE\_temp,char\* device\_id\_temp,char\* sensor\_id\_temp,char\* lon\_temp,char\* lat\_temp,int a)

{

char send\_buf[400]= {0};

char text[100] = {0};

char tmp[25] = {0};

char b[]="0";

char lon\_str\_end[15] = {0};

char lat\_str\_end[15] = {0};

dtostrf(longitudeToOnenetFormat(lon\_temp),3,6, lon\_str\_end); //转换成字符串输出

dtostrf(latitudeToOnenetFormat(lat\_temp),2,6, lat\_str\_end); //转换成字符串输出

//连接服务器

memset(send\_buf, 0, 400); //清空

strcpy(send\_buf, "AT+CIPSTART=\"TCP\",\"");

strcat(send\_buf, OneNetServer);

strcat(send\_buf, "\",80\r\n");

if (sendCommand(send\_buf, "CONNECT", 10000, 5) == Success);

else GPRS\_ERROR(7);

//发送数据

if (sendCommand("AT+CIPSEND\r\n", ">", 3000, 1) == Success);

else GPRS\_ERROR(8);

memset(send\_buf, 0, 400); //清空

/\*准备JSON串\*/

//ARDUINO平台不支持sprintf的double的打印，只能转换到字符串然后打印

if(a==0)

sprintf(text,"{\"datastreams\":[{\"id\":\"%s\",\"datapoints\":[{\"value\":{\"lon\":%s,\"lat\":%s,\"is\":%s}}]}]}",sensor\_id\_temp,lon\_str\_end,lat\_str\_end,b);

else

sprintf(text,"{\"datastreams\":[{\"id\":\"%s\",\"datapoints\":[{\"value\":{\"lon\":%s,\"lat\":%s}}]}]}",sensor\_id\_temp,lon\_str\_end,lat\_str\_end);

/\*准备HTTP报头\*/

send\_buf[0] = 0;

strcat(send\_buf,"POST /devices/");

strcat(send\_buf,device\_id\_temp);

strcat(send\_buf,"/datapoints HTTP/1.1\r\n");//注意后面必须加上\r\n

strcat(send\_buf,"api-key:");

strcat(send\_buf,API\_VALUE\_temp);

strcat(send\_buf,"\r\n");

strcat(send\_buf,"Host:");

strcat(send\_buf,OneNetServer);

strcat(send\_buf,"\r\n");

sprintf(tmp,"Content-Length:%d\r\n\r\n", strlen(text));//计算JSON串长度

strcat(send\_buf,tmp);

strcat(send\_buf,text);

if (sendCommand(send\_buf, send\_buf, 3000, 1) == Success);

else GPRS\_ERROR(9);

char sendCom[2] = {0x1A};

if (sendCommand(sendCom, "\"succ\"}", 3000, 1) == Success);

else GPRS\_ERROR(10);

if (sendCommand("AT+CIPCLOSE\r\n", "CLOSE OK", 3000, 1) == Success);

else GPRS\_ERROR(11);

if (sendCommand("AT+CIPSHUT\r\n", "SHUT OK", 3000, 1) == Success);

else GPRS\_ERROR(11);

}

void initGprs()

{

if (sendCommand("AT\r\n", "OK", 3000, 10) == Success);

else GPRS\_ERROR(1);

if (sendCommand("AT+CREG?\r\n", "OK", 3000, 10) == Success);

else GPRS\_ERROR(2);

if (sendCommand("AT+CGCLASS=\"B\"\r\n", "OK", 3000, 2) == Success);

else GPRS\_ERROR(3);

if (sendCommand("AT+CGDCONT=1,\"IP\",\"CMNET\"\r\n", "OK", 3000, 2) == Success);

else GPRS\_ERROR(4);

if (sendCommand("AT+CGATT=1\r\n", "OK", 3000, 2) == Success);

else GPRS\_ERROR(5);

if (sendCommand("AT+CLPORT=\"TCP\",\"2000\"\r\n", "OK", 3000, 2) == Success);

else GPRS\_ERROR(6);

}

void(\* resetFunc) (void) = 0; //制造重启命令

void GPRS\_ERROR(int num)

{

DebugSerial.print("ERROR");

DebugSerial.println(num);

while (1)

{

digitalWrite(L, HIGH);

delay(300);

digitalWrite(L, LOW);

delay(300);

if (sendCommand("AT\r\n", "OK", 100, 10) == Success)

{

DebugSerial.print("\r\nRESET!!!!!!\r\n");

resetFunc();

}

}

}

unsigned int sendCommand(char \*Command, char \*Response, unsigned long Timeout, unsigned char Retry)

{

clrGprsRxBuffer();

for (unsigned char n = 0; n < Retry; n++)

{

DebugSerial.print("\r\n---------send AT Command:---------\r\n");

DebugSerial.write(Command);

GprsSerail.write(Command);

Time\_Cont = 0;

while (Time\_Cont < Timeout)

{

gprsReadBuffer();

if(strstr(gprsRxBuffer, Response) != NULL)

{

DebugSerial.print("\r\n==========receive AT Command:==========\r\n");

DebugSerial.print(gprsRxBuffer); //输出接收到的信息

clrGprsRxBuffer();

return Success;

}

}

Time\_Cont = 0;

}

DebugSerial.print("\r\n==========receive AT Command:==========\r\n");

DebugSerial.print(gprsRxBuffer);//输出接收到的信息

clrGprsRxBuffer();

return Failure;

}

void Timer1\_handler(void)

{

Time\_Cont++;

}

void gprsReadBuffer() {

while (GprsSerail.available())

{

gprsRxBuffer[gprsBufferCount++] = GprsSerail.read();

if (gprsBufferCount == gprsRxBufferLength)clrGprsRxBuffer();

}

}

void clrGprsRxBuffer(void)

{

memset(gprsRxBuffer, 0, gprsRxBufferLength); //清空

gprsBufferCount = 0;

}

String parseGprmcLat(String s)

{

int pLoc = 0; //paramater location pointer

int lEndLoc = 0; //lat parameter end location

String lat;

if(s.substring(0,4) == "GPRM")

{

//Serial.println(s);

for(int i = 0; i < 5; i++)

{

if(i < 3)

{

pLoc = s.indexOf(',', pLoc+1);

}

if(i == 3)

{

lEndLoc = s.indexOf(',', pLoc+1);

lat = s.substring(pLoc+1, lEndLoc);

}

}

return lat;

}

}

String parseGprmcLon(String s)

{

int pLoc = 0; //paramater location pointer

int lEndLoc = 0; //lat parameter end location

String lon;

if(s.substring(0,4) == "GPRM")

{

//Serial.println(s);

for(int i = 0; i < 7; i++)

{

if(i < 5)

{

pLoc = s.indexOf(',', pLoc+1);

}

if(i == 5)

{

lEndLoc = s.indexOf(',', pLoc+1);

lon = s.substring(pLoc+1, lEndLoc);

}

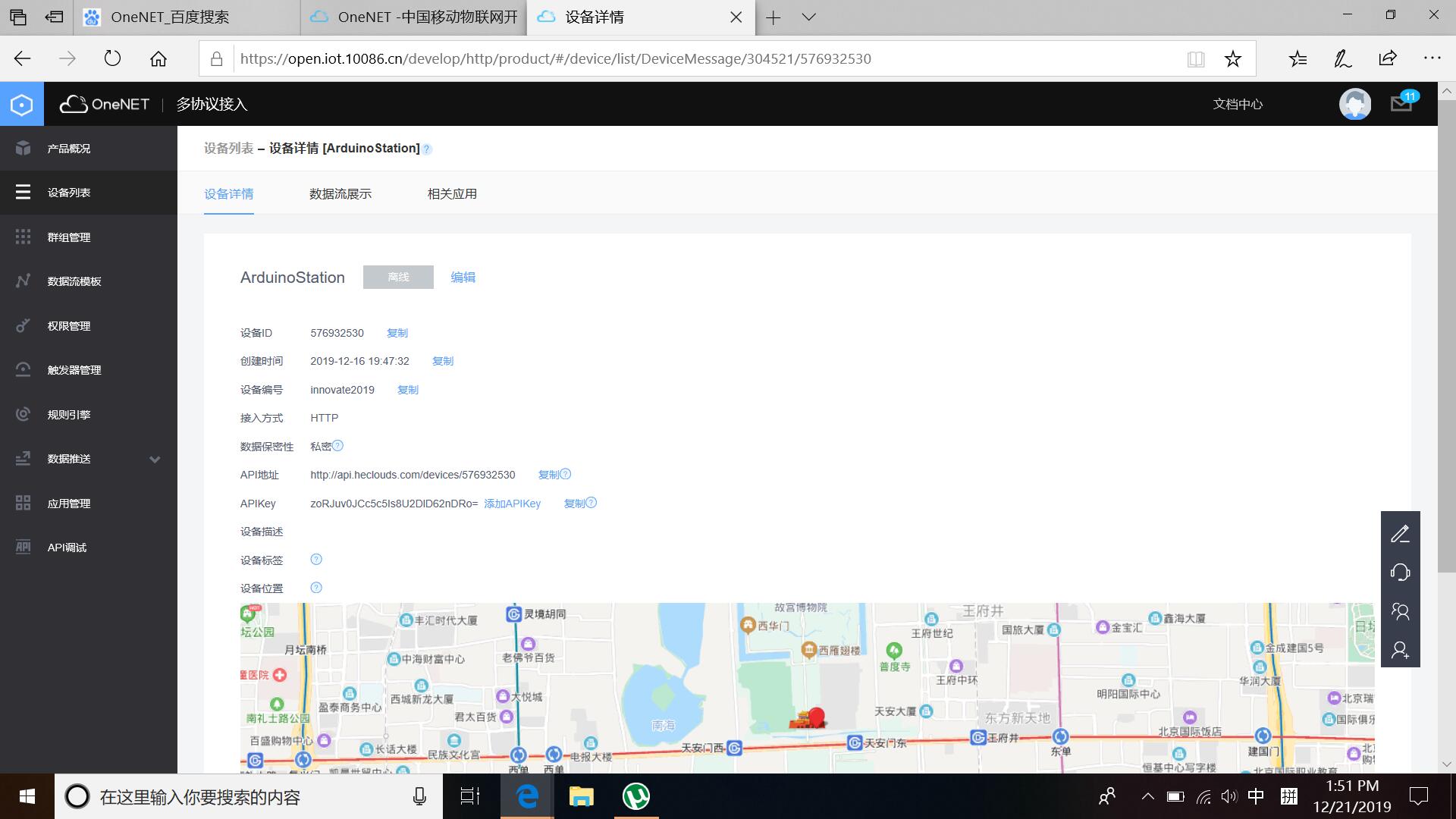
}

return lon;

}

}

**Onenet:**



**微信小程序：（核心代码及其示例）**

<<map id="myMap" style="width: {{mapWidth}}rpx; height: {{mapHeight}}rpx;" latitude="{{lonn1}}" longitude="{{latn1}}" arkers="{{markers}}" show-location bindmarkertap="selectMarket" include-points="{{markers}}" bindmarkertap="toaddress"></map>

1. **小程序获取服务器数据部分：（js)**
2. onLoad: function () {
3. var that = this//不要漏了这句，很重要
4. wx.request({
5. url: 'http://api.heclouds.com/devices/576932530',
6. headers: {
7. 'Content-Type': 'application/json',
8. 'api-key':'zoRJuv0JCc5c5Is8U2DlD62nDRo='
9. },
10. success: function (res) {
11. that.setData({
12. longitude: res.data.longitude,
13. latitude: res.data.latitude,
14. //res代表success函数的事件对，data是固定的，fengxiang是是上面json数据中fengxiang
15. })
16. }
17. })
18. },
19. Api
20. <view class="" hover-class="none" hover-stop-propagation="false" style="width: 100%; height: 100%;">
21. <map id="map" longitude="{{longitude}}" latitude=" {{latitude}}" markers="{{markers}}" scale="13" show-location style="width: 100%; height: 100%;"></map>
22. <view></view>
23. </view>
24. Wxml function
25. page{
26. height: 96%;
27. }
28. Wxss function
29. <view class="root">
30. <image class='background-image' src='./background.jpg' mode="aspectFill"></image>
31. <button class='gps' bindtap="getLocation">
32. <image class='gpsimage' src='./gps1.png'></image>
33. <view>{{locationButton}}</view>
34. </button>
35. </view>
36. Wxml index
37. page{
38. width:100%;
39. height:100%;
40. overflow: hidden;
41. }
42. .header{
43. text-align: center;
44. color: rgb(0, 0, 0);
45. font-size: 23px;
46. padding: 23px 9%;
48. line-height: 1.5;
49. }
50. .gps{
51. position: top;
52. width: 300rpx;
53. height: 300rpx;
54. //margin-top: 20rpx;
55. color: #000000;
56. border-radius: 0rpx;
57. display: flex;
58. flex-direction: column;
59. font-size: 40rpx;
60. background-color: transparent; /\*按钮透明\*/
61. border:none;       /\*设置按钮边框\*/
62. // background-color: #70c7da;
63. }
64. .gpsimage{
65. top: 100px;
66. width: 300rpx;
67. height: 300rpx;
68. }
69. .root {
70. width: 100%;
71. height: 100%;
72. background-color: #70c7da;
73. position: relative;
74. }
75. .background-image{
76. height : 100%;
77. position: absolute;
78. width: 100%;
79. left: 0px;
80. top: 0px;
81. }
82. Wxss index
83. Page({
84. data:{
85. header: "下图是您的家人在的位置",
86. locationButton: "点击获取家人的位置信息"
87. },
88. getLocation: function () {
89. wx.navigateTo({
90. url: '../function/function',
91. })
92. }
93. })
94. Js index
95. Page({
96. data: {
97. header: "欢迎来到盲人拐杖家属客户端",
98. locationButton: "点击按钮进入程式中"
99. },
100. getLocation: function(){
101. wx.navigateTo({
102. url: '../index/index',
103. })
104. }
105. })
106. Start js
107. <view class='header'>{{header}}</view>
108. <button bindtap="getLocation" class="locationButton">{{locationButton}}</button>
109. Start wxml
110. page{
111. background:#80acdfc0;
112. }
113. .header{
114. text-align: center;
115. color: darkgreen;
116. font-size: 40px;
117. padding: 23px 9%;
119. line-height: 1.5;
120. }
121. .locationButton{
122. size: 200px;
123. background-color: rgb(106, 143, 150);
124. color: #3813db;
125. font-size: 25px;
126. top : 45px;
127. }
128. Wxss start
129. {
130. "pages": [
131. "pages/start/start",
132. "pages/index/index",
133. "pages/function/function"
134. ],
135. "window": {
136. "backgroundTextStyle": "light",
137. "navigationBarBackgroundColor": "#fff",
138. "navigationBarTitleText": "盲人拐杖",
139. "navigationBarTextStyle": "black"
140. },
141. "style": "v2",
142. "sitemapLocation": "sitemap.json"
144. }
145. App jason

**展示图片：**











