超声波计时器—下位机部分

超声波模块测距原理：

超声波发射器向某一方向发射超声波，在发射的同时开始计时，超声波在空气中传播，途中碰到障碍物就立即返回来，超声波接收器收到反射波就立即停止计时。声波在空气中的传播速度为340m/s，根据计时器记录的时间t，就可以计算出发射点距障碍物的距离s，即：s=340m/s× t / 2 。这就是所谓的时间差测距法。

下位机计时原理：

运用超声波收发装置，根据测距原理，当没有小车通过时，超声波测定的距离没有发生改变，当小车经过的时候，由于小车车体存在，所以改变了超声波收发的距离，因此距离发生改变的时间点即为小车经过的时间点。

工作流程：

当小车经过时，测得超声波收发距离发生改变，记录该事件点并发送“passed“给上位机，开始计时，当小车再次经过时，同理，将测得时间点，并发送”arrived“给上位机。如果在小车出发之前发生误触，则由上位机发送指令”r“,重新开始计时，在小车出发后发生误触，上位机发出指令”e“，继续计时，并按照下位机发送给上位机的第一次”passed“和第二次”arrived“计时。

状态机：（当系统处于某状态（S1）时，如果发生了什么事情(E)，就执行某功能(F)，然后系统变成新状态（S2））

在本装置执行计时任务时可设定以下两种状态：

状态1：prepare 下位机处于预备状态，当小车经过时，发送“passed“给上位机。

状态2：timing 处于计时状态，

在比赛中主要有以下3种情况：

1. 当小车正常出发，状态1停止，下位机处于状态2计时，小车到达终点，状态1停止，回到状态1，计时结束等待下次计时。
2. 当小车出发前发生误触，状态1停止并触发状态2，此时上位机发出指令“r“，令下位机恢复状态1，从而重新计时（在按下“重置”按钮的同时上位机串口发送“r”）
3. 当小车出发后发生误触，状态2停止并回到状态，上位机发出指令“e“，使之恢复状态2，计时继续。（为了实现这个计时继续，所以就不能让上位机显示的时间停止，而是让时间一直走，只需记录下一次接收到“arrive”的时间点，按下“误触”按钮，上位机串口发送“e”）

代码：

const int TrigPin = 2;

const int EchoPin = 3;

float cm;

long last\_time = 0;

boolean pass = false;

boolean prepare =true ;

boolean timing = false;

boolean is\_car\_pass(void);

char val;

boolean check\_passed();

char getRawVal();

void setup()

{

Serial.begin(9600);

pinMode(TrigPin, OUTPUT);

pinMode(EchoPin, INPUT);

}

void loop()

{

if (prepare == 1)

{

if (is\_car\_pass())

{

Serial.print"passed");

delay(1000);

timing = true;

prepare = false;

}

}

if (timing == true)

{

if (is\_car\_pass())

{

Serial.print("arrived");

delay(1000);

prepare = true;

timing = false;

}

}

if (Serial.available() > 0)

{

val = Serial.read();

if (val == 'e')

{

timing = true;

prepare = false;

}

if (val == 'r')

{

timing = false;

prepare = true;

}

}

}

boolean is\_car\_pass(void)

{

if (check\_passed())

{

pass = true;

return 1;

}

else if (pass == true)

{

pass = false;

}

}

boolean check\_passed()

{

if(getRawVal() < 10)

{

delay(15);

if(getRawVal() < 10)

{

return true;

}

}

return false;

}

char getRawVal()

{

digitalWrite(TrigPin, LOW); //低高低电平发一个短时间脉冲去TrigPin

delayMicroseconds(2);

digitalWrite(TrigPin, HIGH);

delayMicroseconds(10);

digitalWrite(TrigPin, LOW);

return cm = pulseIn(EchoPin, HIGH) / 58; //将回波时间换算成cm

}