



红外测速仪

基于人工智能与红外光电开关的自动测速仪





目录

CONTENTS

01 作品设计思路与计划

02 作品创新点

03 材料清单

04 制作过程

05 成果呈现



作品设计思路与计划

01

◆ 作品背景 ◆ 系统构成与设计要求 ◆ 应用场景



作品背景

速度，一个与我们的日常生活息息相关的物理量，小到上学时的步行速度大到奥林匹克运动会上的百米竞速，都是速度的表现方式。瞬时速度公式在物理学和工程学中的应用非常广泛。例如，在汽车制造业中，我们需要知道车辆在某个时刻的速度，这样我们才能更好地改进汽车的设计。此外，在工程领域，工程师需要知道机器在某个时刻的速度，这样才能更好地维护和保养机器。

微积分在实际中的应用非常广泛，它可以用于解决各种问题。例如，在金融领域，我们可以用微积分来计算股票的收益率和波动率。在医学领域，我们可以用微积分来计算心脏的工作以及药物在人体内的代谢过程。此外，在地球科学领域，微积分可以用于计算地球上的高度或者海平面的变化。

在一次咨询调查中，我偶然听一名被调查同学提到了瞬时速度，这一点令我回忆起利用瞬时速度解析微积分。这使我想利用电子设备设计一个实验，用以验证微积分对瞬时速度的解析，及瞬时速度的公式

作品设计

- ◆ 微积分是数学的一个重要分支，包括微分 (Differentiation) 和积分 (Integration) 两个部分。内容主要包括极限、微分学、积分学及其应用。微分用于研究函数在某一点处的变化率，而积分则用于求出函数所占据的面积或者体积。微积分的核心思想是将一个复杂的问题分解成许多小问题，并通过处理这些小问题来得出答案。其中，微分部分的一个重要应用就是瞬时速度公式。
- ◆ 微积分中的瞬时速度公式表达式为： $v(t) = \lim_{h \rightarrow 0} [f(t+h) - f(t)]/h$ 。其中， $v(t)$ 表示在时间 t 时刻的瞬时速度， $f(t)$ 表示一个物体在时间 t 时刻的位置。这个公式告诉我们，如果我们知道一个物体在某一个时刻的位置，那么我们就能够通过微积分的概念来计算它在该时刻的瞬时速度。
- ◆ 假设一段物体进行匀加速直线运动的位移，利用微积分思想，将这一段路程微分为无数段的小路程，单一研究其中一段。此时，该段路程的瞬时速度加上加速度再乘与时间，即为该段路程的长度。此时这段小路程是一段路程微分为无数小路程后取其中一段，所以时间也被微分至极小，直至此时加速度的影响极小，可忽略。将此时的平均速度视为瞬时速度。
- ◆ 为验证这一猜想，设计一个实验。在斜面上安装四个光电开关，将斜面分为等距的几段路程。光电开关测小车的遮光时间作为，将小车自静止放下，如果满足第一个光电开关测得 V_1^2 比第二个光电开关测得 V_2^2 比第三个光电开关测得 V_3^2 为 1: 2: 3。则这一瞬时速度表示公式可行。

系统构成



设计要求

- ◆ 作品坚固、美观
- ◆ 操作方便、简单
- ◆ 测量数据准确

作品应用场景



中学生物理实验室

用于中学生有关数学微积分及物理
瞬时速度的猜想验证以及实验探究

作品创新点

02



作品创新点



结合实际

与生活实际相结合
教学容易理解



操作简单

传感器自动测量数据
操作简单



教学方便

数据会显示在掌控
板屏幕上
学生学习容易



结果清晰

实验进行对比
结果容易得出

材料清单

03



材料清单

序号	设备	连接引脚	数据	备注	数量
1	掌控板2.0		长度	主控	1
2	I/O扩展板			连接外置传感器	1
3	M18漫反射 红外线可调 距光电开关	P13、P15	时间	自动测量	2
4	其他材料：5mm椴木板、若干杜邦线				
合计					

制作过程

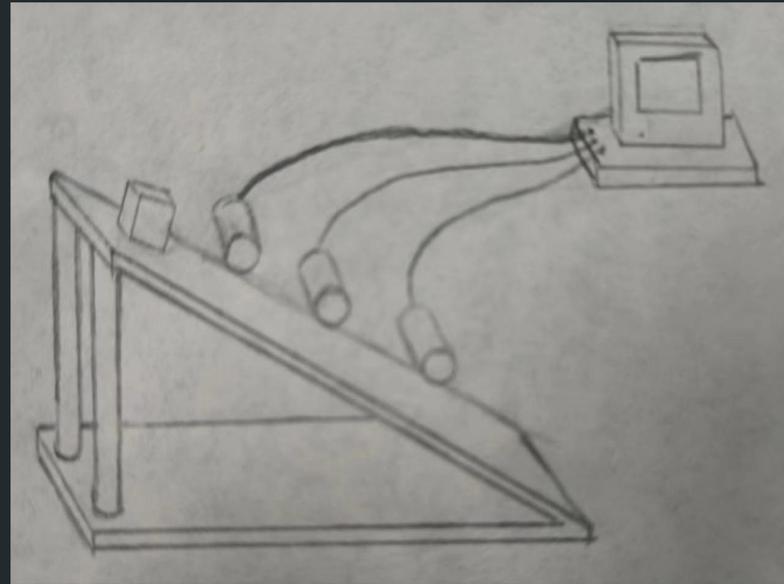
04



一、初步构思

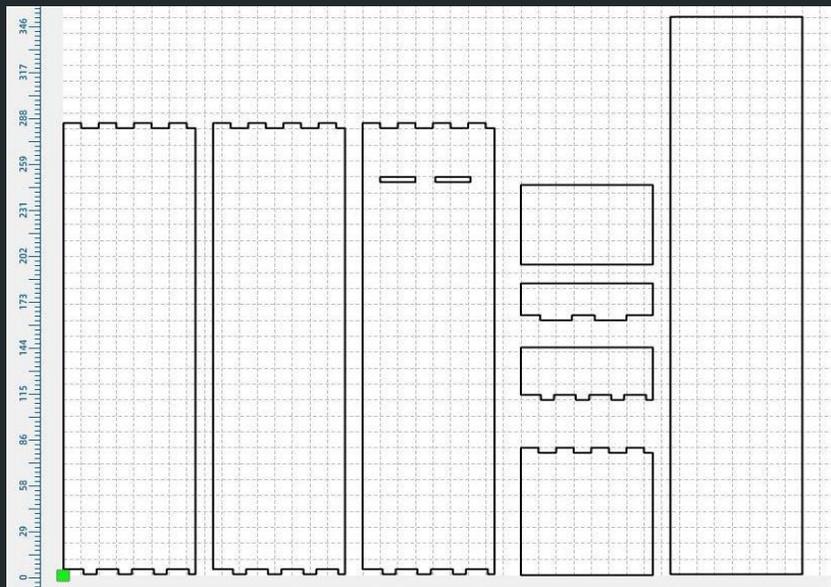


咨询调研

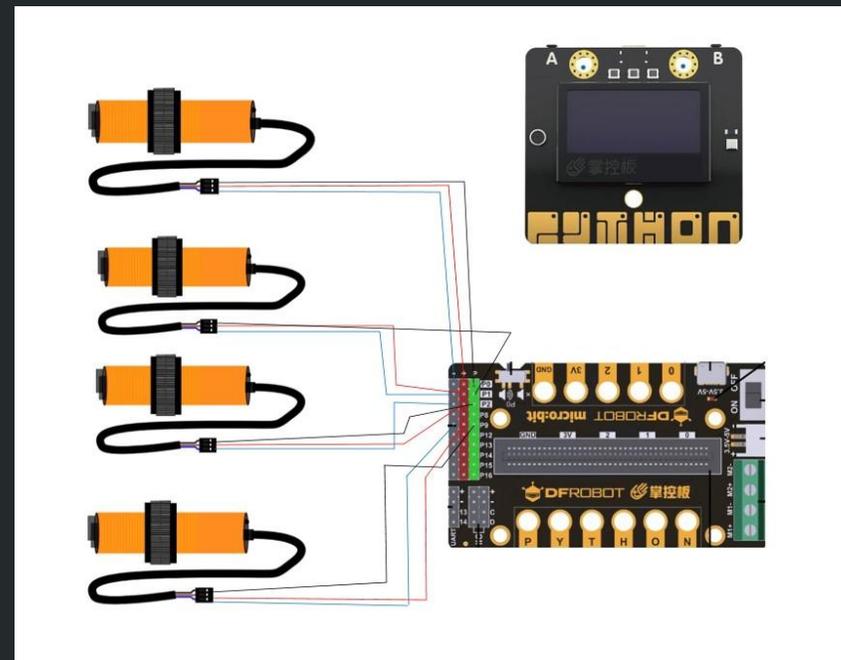


草图设计

二、作品设计



激光切割



硬件连接

三、模型制造



四、编程设计

掌控版图形化编程

ESP32 主程序

- 设置 A 的值为 0
- 屏幕显示文字 "光电测速仪" 在坐标 X: 80 Y: 5 预览
- 屏幕显示文字 "按下A键开始测速" 在坐标 X: 15 Y: 22 预览
- 等待 1 秒
- 循环执行
 - 如果 按钮 A 被按下? 那么执行
 - 等待 0.2 秒
 - 将 A 增加 1
 - 如果 变量 A 除以 2 的余数 = 1 那么执行
 - 设置 V1 的值为 变量长度 / 变量 t2 - 变量 t1
 - 设置 V2 的值为 变量长度 / 变量 t4 - 变量 t3
 - 设置 V3 的值为 变量长度 / 变量 t6 - 变量 t5
 - 设置 V4 的值为 变量长度 / 变量 t8 - 变量 t7
 - 屏幕显示文字 合并 "V1=" 合并 变量 V1 "m/s" 在第 1 行
 - 屏幕显示文字 合并 "V2=" 合并 变量 V2 "m/s" 在第 2 行
 - 屏幕显示文字 合并 "V3=" 合并 变量 V3 "m/s" 在第 3 行
 - 屏幕显示文字 合并 "V4=" 合并 变量 V4 "m/s" 在第 4 行
 - 如果 变量 A 除以 2 的余数 = 0 那么执行
 - T1
 - T2
 - T3
 - T4
 - x
 - y

定义 T1

- 如果 读取数字引脚 P0 = 0 那么执行
 - 设置 t1 的值为 系统运行时间(ms)
 - 重复执行直到 读取数字引脚 P0 = 1
 - 设置 t2 的值为 系统运行时间(ms)
- 屏幕显示文字 合并 "T1=" 合并 变量 t2 - 变量 t1 "ms" 在第 1 行

定义 T2

- 如果 读取数字引脚 P1 = 0 那么执行
 - 设置 t3 的值为 系统运行时间(ms)
 - 重复执行直到 读取数字引脚 P1 = 1
 - 设置 t4 的值为 系统运行时间(ms)
- 屏幕显示文字 合并 "T2=" 合并 变量 t4 - 变量 t3 "ms" 在第 2 行

定义 T3

- 如果 读取数字引脚 P2 = 0 那么执行
 - 设置 t5 的值为 系统运行时间(ms)
 - 重复执行直到 读取数字引脚 P2 = 1
 - 设置 t6 的值为 系统运行时间(ms)
- 屏幕显示文字 合并 "T3=" 合并 变量 t6 - 变量 t5 "ms" 在第 3 行

定义 x

- 如果 0 < 1 那么执行
 - 屏幕显示文字 "长度" 在坐标 X: 90 Y: 0 预览
 - 屏幕显示文字 变量长度 在坐标 X: 90 Y: 15 预览
 - 屏幕显示文字 "mm" 在坐标 X: 90 Y: 30 预览
 - 等待 0.2 秒

定义 T2

- 当触摸按钮 N 被按下
 - 将 长度 增加 -10
- 当触摸按钮 O 被按下
 - 将 长度 增加 -5
- 当触摸按钮 H 被按下
 - 将 长度 增加 -0.1
- 当触摸按钮 T 被按下
 - 将 长度 增加 0.1
- 当触摸按钮 Y 被按下
 - 将 长度 增加 1
- 当触摸按钮 P 被按下
 - 将 长度 增加 10

定义 T4

- 如果 读取数字引脚 P9 = 0 那么执行
 - 设置 t7 的值为 系统运行时间(ms)
 - 重复执行直到 读取数字引脚 P9 = 1
 - 设置 t8 的值为 系统运行时间(ms)
- 屏幕显示文字 合并 "T4=" 合并 变量 t8 - 变量 t7 "ms" 在第 4 行

定义 y

- 如果 按钮 B 被按下? 那么执行
 - 设置 长度 的值为 0
 - 等待 0.2 秒

掌控板代码编程

```
4 *
5 */
6 #include <MPython.h>
7
8 // 动态变量
9 volatile float mind_n_t1, mind_n_t2, mind_n_A, mind_n_ChangDu, mind_n_V1, mind_n_t4,
10             mind_n_t3, mind_n_V2, mind_n_t6, mind_n_t5, mind_n_V3, mind_n_t8,
11             mind_n_t7, mind_n_V4;
12 // 函数声明
13 void DF_T1();
14 void DF_T3();
15 void DF_x();
16 void DF_T2();
17 void pin4TouchCallback();
18 void pin15TouchCallback();
19 void DF_T4();
20 void pin13TouchCallback();
21 void pin12TouchCallback();
22 void pin14TouchCallback();
23 void DF_y();
24 void pin27TouchCallback();
25
26
27 // 主程序开始
28 void setup() {
29     mPython.begin();
30     touchPadN.setTouchedCallback(pin4TouchCallback);
31     touchPadO.setTouchedCallback(pin15TouchCallback);
32     touchPadH.setTouchedCallback(pin13TouchCallback);
33     touchPadT.setTouchedCallback(pin12TouchCallback);
34     touchPadY.setTouchedCallback(pin14TouchCallback);
35     touchPadP.setTouchedCallback(pin27TouchCallback);
36     mind_n_A = 0;
37     display.setCursor(30, 5);
38     display.print("光电测速仪");
39     display.setCursor(15, 22);
40     display.print("按下A键开始测速");
```

```
41     delay(1000);
42 }
43 void loop() {
44     if ((buttonA.isPressed())) {
45         delay(200);
46         mind_n_A += 1;
47     }
48     if (((int32_t(mind_n_A) % (int32_t(2)))==1)) {
49         mind_n_V1 = (mind_n_ChangDu / (mind_n_t2 - mind_n_t1));
50         mind_n_V2 = (mind_n_ChangDu / (mind_n_t4 - mind_n_t3));
51         mind_n_V3 = (mind_n_ChangDu / (mind_n_t6 - mind_n_t5));
52         mind_n_V4 = (mind_n_ChangDu / (mind_n_t8 - mind_n_t7));
53         display.setCursorLine(1);
54         display.println((String("V1=") + String((String(mind_n_V1) + String("m/s")))));
55         display.setCursorLine(2);
56         display.println((String("V2=") + String((String(mind_n_V2) + String("m/s")))));
57         display.setCursorLine(3);
58         display.println((String("V3=") + String((String(mind_n_V3) + String("m/s")))));
59         display.setCursorLine(4);
60         display.println((String("V4=") + String((String(mind_n_V4) + String("m/s")))));
61     }
62     if (((int32_t(mind_n_A) % (int32_t(2)))==0)) {
63         DF_T1();
64         DF_T2();
65         DF_T3();
66         DF_T4();
67         DF_x();
68         DF_y();
69     }
70 }
71
72
73 // 自定义函数
74 void DF_T1() {
75     if (((digitalRead(P0))==0)) {
76         mind_n_t1 = millis();
77         while (!((digitalRead(P0))==1)) {
```

掌控板代码编程

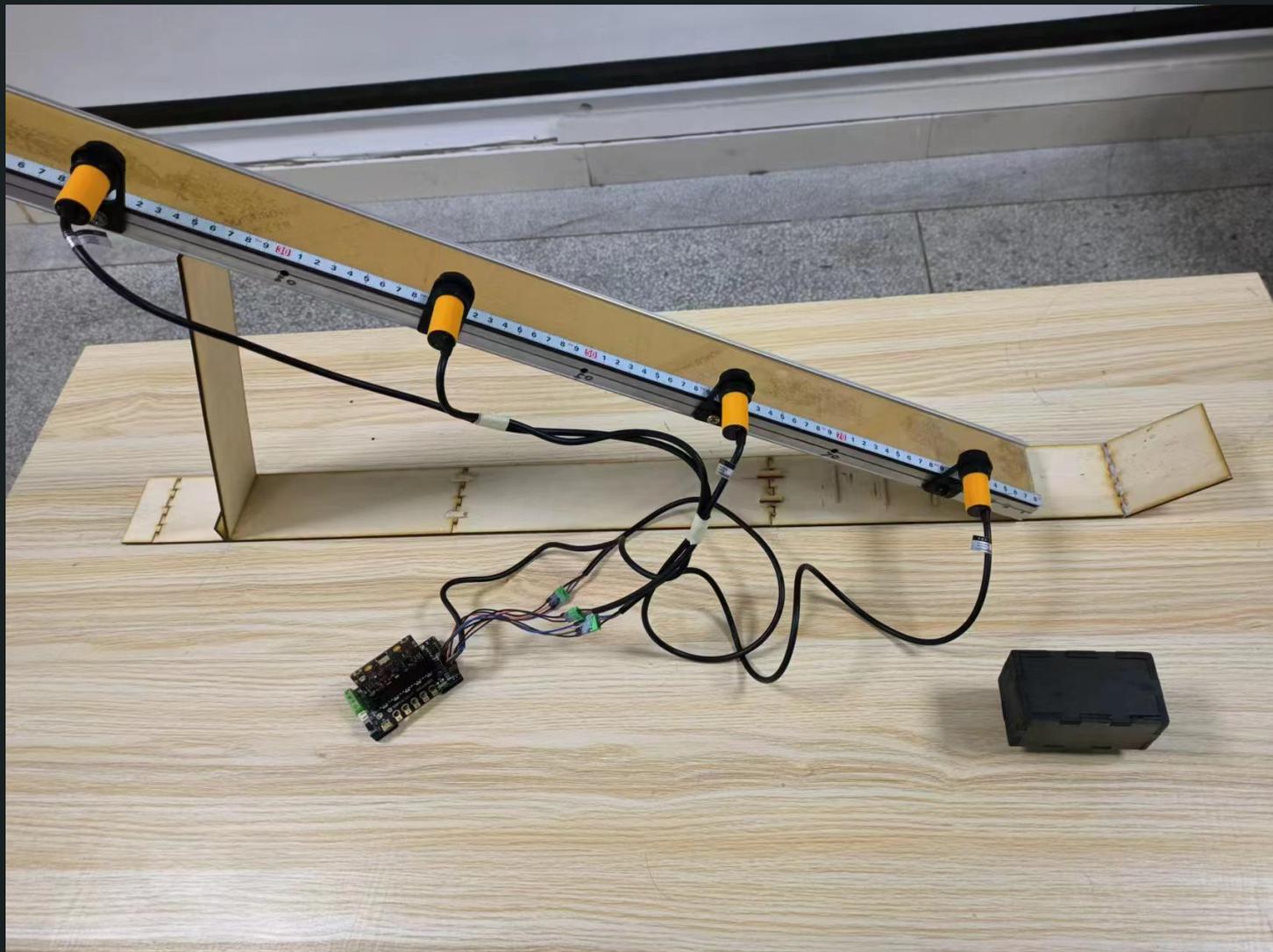
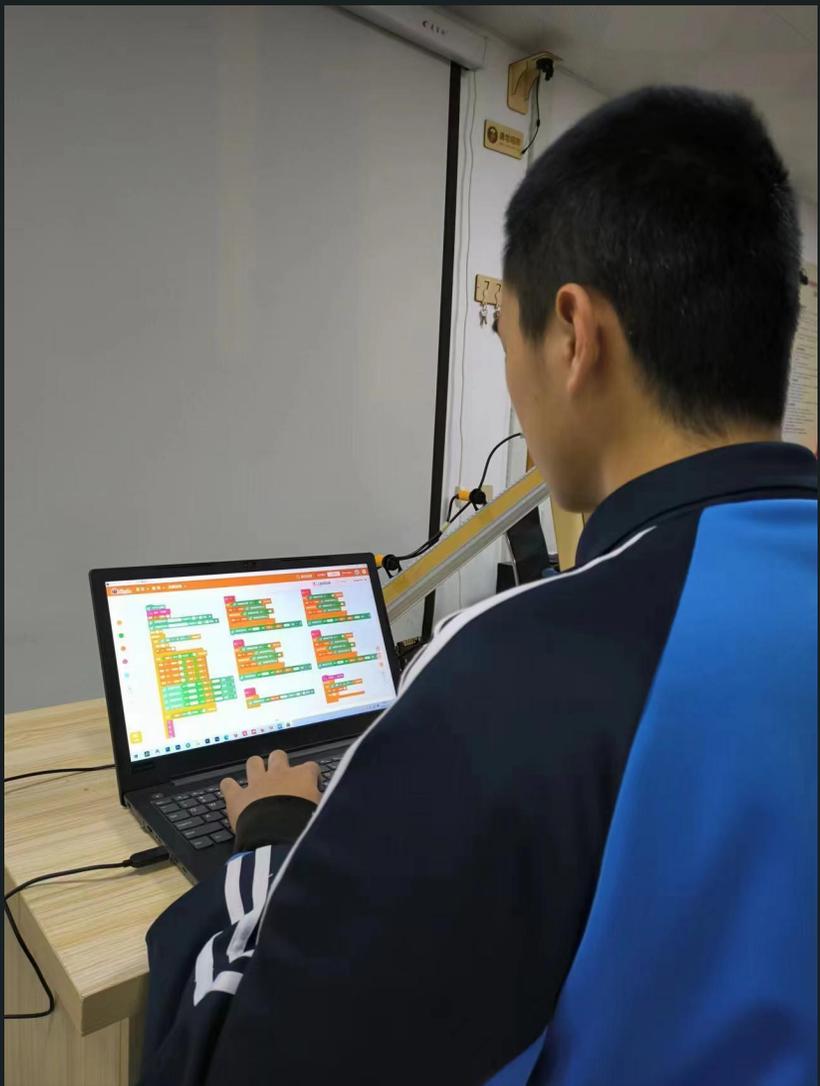
```
77 while (!((digitalRead(P0))==1)) {
78     mind_n_t2 = millis();
79     yield();
80 }
81 }
82 display.setCursorLine(1);
83 display.println((String("T1=") + String((String((mind_n_t2 - mind_n_t1)) + String("ms")))));
84 }
85 void DF_T3() {
86     if (((digitalRead(P2))==0)) {
87         mind_n_t5 = millis();
88         while (!((digitalRead(P2))==1)) {
89             mind_n_t6 = millis();
90             yield();
91         }
92     }
93     display.setCursorLine(3);
94     display.println((String("T3=") + String((String((mind_n_t6 - mind_n_t5)) + String("ms")))));
95 }
96 void DF_x() {
97     if ((0<1)) {
98         display.setCursor(90, 0);
99         display.print("长度");
100        display.setCursor(90, 15);
101        display.print(mind_n_ChangDu);
102        display.setCursor(90, 30);
103        display.print("mm");
104        delay(200);
105    }
106 }
107 void DF_T2() {
108     if (((digitalRead(P1))==0)) {
109         mind_n_t3 = millis();
110         while (!((digitalRead(P1))==1)) {
111             mind_n_t4 = millis();
112             yield();
113         }

```

```
114     }
115     display.setCursorLine(2);
116     display.println((String("T2=") + String((String((mind_n_t4 - mind_n_t3)) + String("ms")))));
117 }
118 void DF_T4() {
119     if (((digitalRead(P9))==0)) {
120         mind_n_t7 = millis();
121         while (!((digitalRead(P9))==1)) {
122             mind_n_t8 = millis();
123             yield();
124         }
125     }
126     display.setCursorLine(4);
127     display.println((String("T4=") + String((String((mind_n_t8 - mind_n_t7)) + String("ms")))));
128 }
129 void DF_y() {
130     if ((buttonB.isPressed())) {
131         mind_n_ChangDu = 0;
132         delay(200);
133     }
134 }
135
136 // 事件回调函数
137 void pin4TouchCallback() {
138     mind_n_ChangDu -= 10;
139 }
140 void pin15TouchCallback() {
141     mind_n_ChangDu -= 1;
142 }
143 void pin13TouchCallback() {
144     mind_n_ChangDu -= 0.1;
145 }
146 void pin12TouchCallback() {
147     mind_n_ChangDu += 0.1;
148 }
149 void pin14TouchCallback() {
150     mind_n_ChangDu += 1;

```

五、成品调试



成果呈现

05



实验及数据：[16dfda679333e6a1d54a1d68fd33faa2.mp4](#)

序号	高度1		高度2	
	1	2	1	2
速度1	1.25m/s	1.23m/s	3.46m/s	3.40m/s
速度2	1.80m/s	1.82m/s	4.90m/s	4.94m/s
速度3	2.17m/s	2.21m/s	5.99m/s	6.02m/s

作品相片



创作总结

◆ 不足之处

- ◆ 本设计使用的是用作教学的独立智能硬件，成本较高，如果需要产品化设计，可以使用价格较低的普通硬件。
- ◆ 目前这一装置可进行的实验较少，可以可以考虑从红外的特点进行特化性改造，使之可以进行更多的中学实验。
- ◆ 实体采用激光切割的木板，外形不够美观，可以考虑使用塑胶等材质制作外壳。
- ◆ 光电开关之间存在间距等，测量计算存在误差。

◆ 心得体会

创造离不开实践，当前情况下我的大部分知识与经验都来自于前人的实践。然而“实践出真知”，只有切身经历了创造实践这一过程才能真正发现问题所在，积累经验。

例如：在创作过程中，代码的编辑，模型的设计或多或少都存在一些小问题，但在指导老师的帮助下，都妥善解决了。经过这一次实践，我学习到了许多课上未教的知识，培养了更坚毅的内心，以及坚定了为科学献身的决心。



2023

谢谢聆听

感谢老师的指导

