



Week 2 Study Note

Arduino学习

pin一般命名input, led一般命名outputd

```
void setup(){
    pinMode(I/O接口, OUTPUT);
    pinMode(I/O接口, INPUT);
}

void loop(){
    digitalWrite(inpin)
```

`digitalRead (I/O接口) ;` //表示读取某个接口的电平值

`digitalWrite(I/O接口, 电平 (1/0或者HIGH/LOW));` // 表示将某个接口改成某个电平

调制模拟信号就是通过在一段时间内来回切换高低电平从而控制这段时间内高低电平出现的时间比例，高电平出现的时间比例越多那么输出电压就越接近5V高电平，低电平出现的时间比例越多输出电压就越接近0V低电平。

而这个所谓的时间比例就是所谓的“**占空比**”。

占空比实际上只能获得255的等分比上的电压值，原因是：计算器只保留8bit。

模拟接口输出的是CMOS信号而非数字接口那种TTL信号

模拟接口里面的模数转换器用的是一个10位二进制空间。例如，我们输入一个5V的电平，那么模数转换器会自动把0对应0V，1023对应5V，然后我们可以算出 $5V / 1023 = 0.00488V = 4.8mV$ ，所以1对应4.8mV, 2对应9.6mV, 以此类推一直到5V。

arduino可接受的最大电压是5V，高了会烧板子

```
val = analogRead(potpin); //读取传感器的模拟值并赋给val（即将模拟端口的电平给val）
```

波特率也称调制速率是每秒传输的二进制数据数目，速率是bps，同波特率才可以实现串口通信

每秒通过信道传输的码元数（码元传输速率），通过不同的调制方法可以在一个码元上负载多个比特信息 数据信号对载波的调制速率

“波特率”表示每秒种传输离散信号事件的个数，或每秒信号电平的变化次数，单位为band（波特）。即波特率所表示的是调制速度，是单位时间内传输线路上调制状态的变化数。

调制后同一时间里所包含的数据数量，波特率不同的话数据采集会出错，在单片机中若晶振采集速度不够的话也会出现数据错误

```
Serial.begin(); //开启串行通信接口并设置通信波特率
```

小车电控遇见的问题以及初步解决方案

后续想要实现遥控器（手柄）控制电机的转速，需要用到H桥、控制电路以及升压器。目前组内有很多低内阻（0.1欧）直流电机无法使用，原因是供电充电宝电压太低导致电流过大，直流电机无法带动。后续需要升压模块进行升压。

回答SA上课时的一些问题

Question 1 - 第一个指令中9600代表波特率。请弄清楚波特率是什么。

```
void setup(){
  Serial.begin(9600);
  Serial.println("Hello, SDIM!");
}
```

- 串行通信不是自计时的（不同于SPI或I2C等），因此发送方和接收方都必须就时钟速率达成共识。

波特率是串口通信时的一个参数，决定信息传输的速率。9600意味着每秒传输9600个二进制位数、每个数据字节=10二进制，所以等于960字节/秒。由于我们默认一般都是

9600bps，所以需要告诉Arduino需要以9600的速度来接受和上传数据，这样跟电脑沟通才不会出现乱码。

Question 2 - 尝试逐行解释代码的含义。

```
#define LED 13//定义变量
void setup() { //开始时运行一次的函数
  pinMode(LED, OUTPUT); //定义13为输入端口
}
void loop() { //程序执行一直循环的主函数
  digitalWrite(LED, HIGH); //设置LED端口为高电平1
  delay(50000); //延时50000ms
  digitalWrite(LED, LOW); //设置LED端口为低电平0
  delay(1000); //延时1000ms
}
```

Question 3:-

了解蜂鸣器的分类以及不同类型蜂鸣器的工作原理

包括有源蜂鸣器和无源蜂鸣器。有源蜂鸣器中内置电源，需要外接5v电压，要注意正负极

找到或者自己制作一段旋律，用Arduino与蜂鸣器演奏

这个受我的音乐水平限制，只好去网上找了一段旋律



音调 音符	1	2	3	4	5	6	7
A	221	248	278	294	330	371	416
B	248	278	294	330	371	416	467
C	131	147	165	175	196	221	248
D	147	165	175	196	221	248	278
E	165	175	196	221	248	278	312
F	175	196	221	234	262	294	330
G	196	221	234	262	294	330	371

音调 音符	1	2	3	4	5	6	7
A	441	495	556	589	661	742	833
B	495	556	624	661	742	833	935
C	262	294	330	350	393	441	495
D	294	330	350	393	441	495	556
E	330	350	393	441	495	556	624
F	350	393	441	495	556	624	661
G	393	441	495	556	624	661	742

音调 音符	1	2	3	4	5	6	7
A	882	990	1112	1178	1322	1484	1665
B	990	1112	1178	1322	1484	1665	1869
C	525	589	661	700	786	882	990
D	589	661	700	786	882	990	1112
E	661	700	786	882	990	1112	1248
F	700	786	882	935	1049	1178	1322
G	786	882	990	1049	1178	1322	1484

```
#define NTD0 -1
#define NTD1 294
#define NTD2 330
#define NTD3 350
#define NTD4 393
#define NTD5 441
#define NTD6 495
#define NTD7 556
```

```
#define NTDL1 147
#define NTDL2 165
```

```
#define NTDL3 175
#define NTDL4 196
#define NTDL5 221
#define NTDL6 248
#define NTDL7 278
```

```
#define NTDH1 589
#define NTDH2 661
#define NTDH3 700
#define NTDH4 786
#define NTDH5 882
#define NTDH6 990
#define NTDH7 112
//列出全部D调的频率
#define WHOLE 1
#define HALF 0.5
#define QUARTER 0.25
#define EIGHTH 0.25
#define SIXTEENTH 0.625
//列出所有节拍
int tune[]= //根据简谱列出各频率
{
NTD3,NTD3,NTD4,NTD5,
NTD5,NTD4,NTD3,NTD2,
NTD1,NTD1,NTD2,NTD3,
NTD3,NTD2,NTD2,
NTD3,NTD3,NTD4,NTD5,
NTD5,NTD4,NTD3,NTD2,
NTD1,NTD1,NTD2,NTD3,
NTD2,NTD1,NTD1,
NTD2,NTD2,NTD3,NTD1,
NTD2,NTD3,NTD4,NTD3,NTD1,
NTD2,NTD3,NTD4,NTD3,NTD2,
NTD1,NTD2,NTDL5,NTD0,
NTD3,NTD3,NTD4,NTD5,
NTD5,NTD4,NTD3,NTD4,NTD2,
NTD1,NTD1,NTD2,NTD3,
NTD2,NTD1,NTD1
};
float durt[]= //根据简谱列出各节拍
{
1,1,1,1,
1,1,1,1,
1,1,1,1,
1+0.5,0.5,1+1,
1,1,1,1,
1,1,1,1,
1,1,1,1,
1+0.5,0.5,1+1,
1,1,1,1,
1,0.5,0.5,1,1,
1,0.5,0.5,1,1,
1,1,1,1,
```

```

1,1,1,1,
1,1,1,0.5,0.5,
1,1,1,1,
1+0.5,0.5,1+1,
};
int length;
int tonepin=8;    //得用6号接口
void setup()
{
  pinMode(tonepin,OUTPUT);
  length=sizeof(tune)/sizeof(tune[0]);    //计算长度
}
void loop()
{
  for(int x=0;x<length;x++)
  {
    tone(tonepin,tune[x]);
    delay(500*durt[x]);    //这里用来根据节拍调节延时，500这个指数可以自己调整，在该音乐中，我发现用500比较合适。
    noTone(tonepin);
  }
  delay(2000);
}

```

Question 附加题1

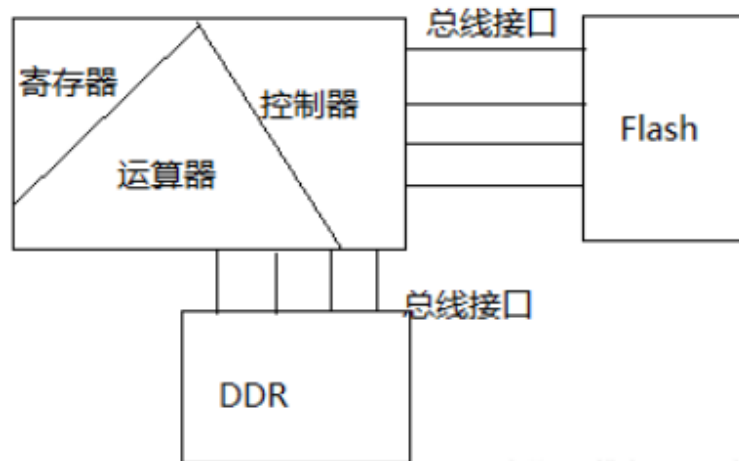
我们经常描述嵌入式设备上电，请问“上电”的含义是什么？

给设备通电，让cpu开始运行，整个系统开始运行

了解编译器的任务。

编译器是将高级语言（C，C++，JAVA之类）变成中间语言的程序，相当于一种能够让机器能读懂人的代码的程序。我们Arduino使用的编译器是gcc，可以转化多种语言。

尝试理解并绘制Arduino代码从编写到最终可以在Arduino上执行的流程图。



首先，程序以机器码的形式，即二进制码的形式存在FLASH中；
上电后，CPU通过控制器将待运行的程序从FLASH中读入内存中；
代码在内存中运行时，控制器将需要计算的数据存入寄存器中；
运算器从寄存器中读取数据进行运算，并将结果存入寄存器中；
控制器将寄存器中的结果读入内存中；
由此，形成一个闭环的程序运行过程。

版权声明：本文为CSDN博主「Clearfix_Xia」的原创文章，遵循CC 4.0 BY-SA版权协议，
转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/csdnxmlj/article/details/103809310>

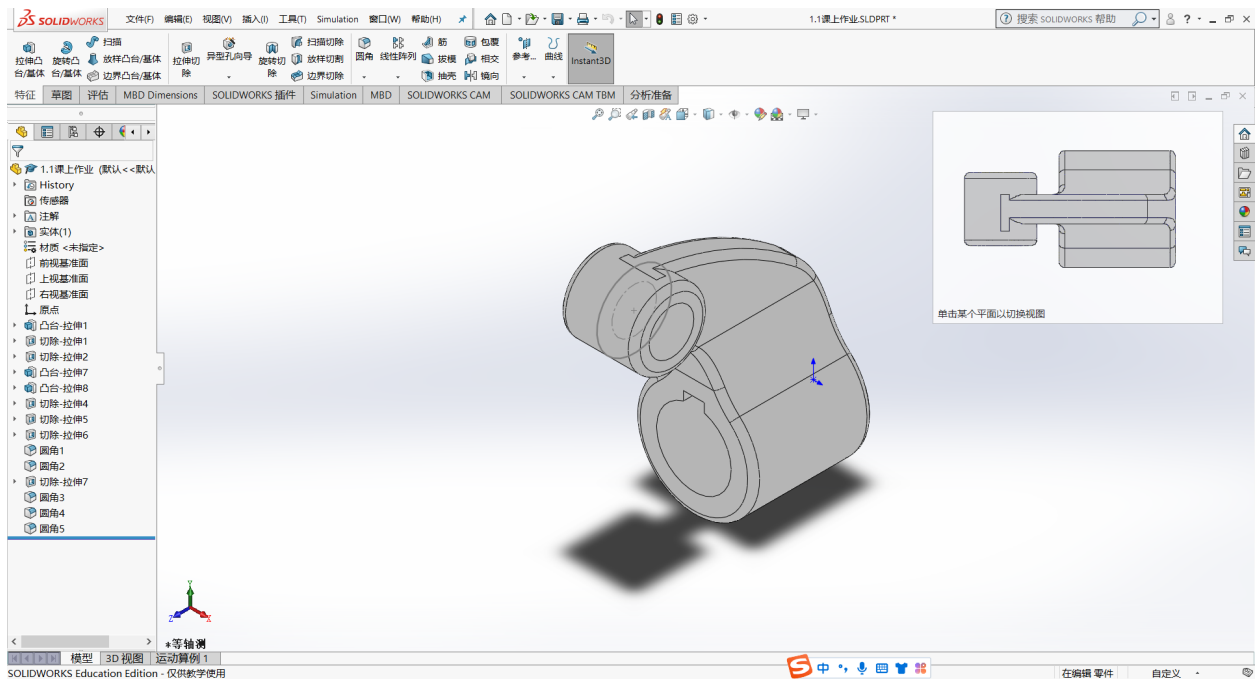
Question 附加题2

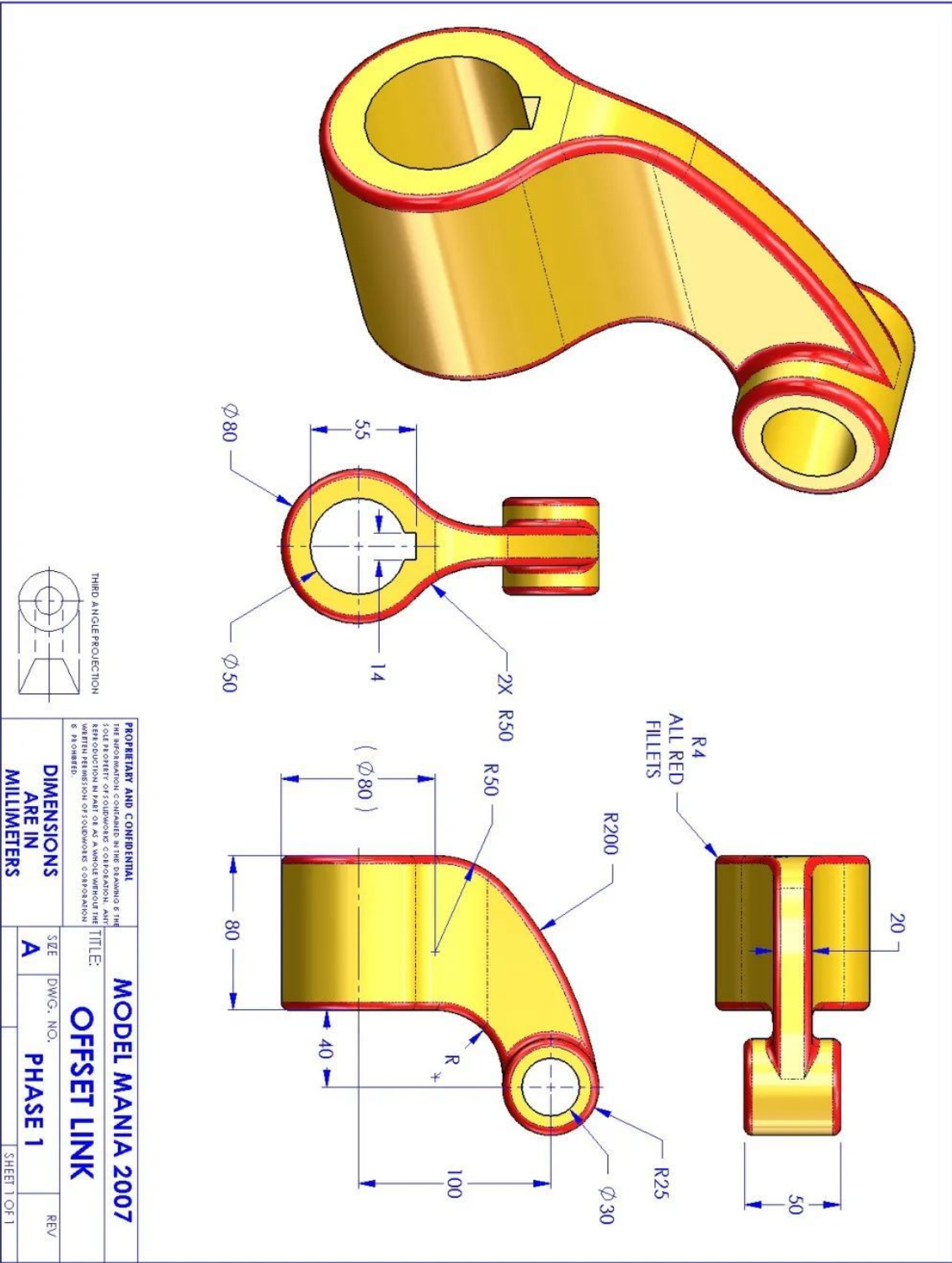
关于PWM的学习资料

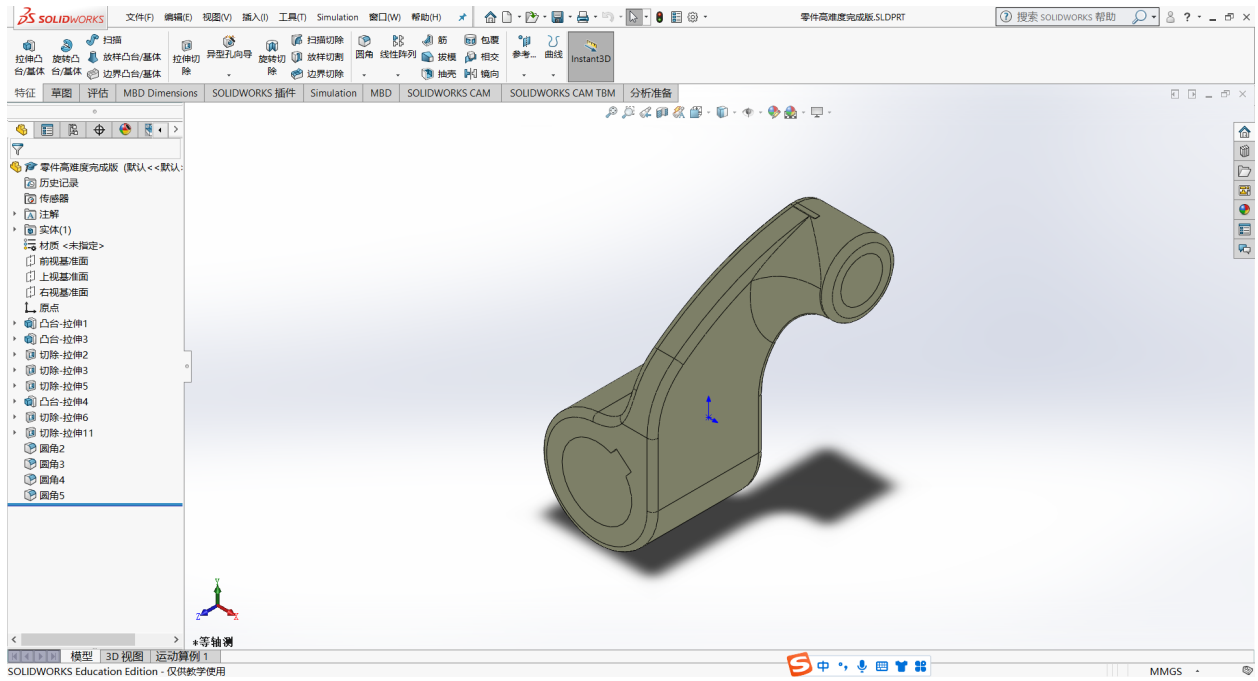
PWM就是通过调整直流电压占空比，来实现调节电压值的效果。

SW学习

本周老师在课上交了我们如何使用Solidworks 3D绘图软件，他可以让我们很好的制作零部件进行3D打印，是个很方便使用的工具。在完成课上任务后，我在课下完成了两幅图的绘制。不得不说，软件虽然看起来很麻烦，但是实际上手操作了两次之后，就感觉大概学会了基本的使用方法。**我画零件的大概流程是：先将零件的某个特征画出来，再讲零件的整个体粗略画出来，然后慢慢抠出细节**

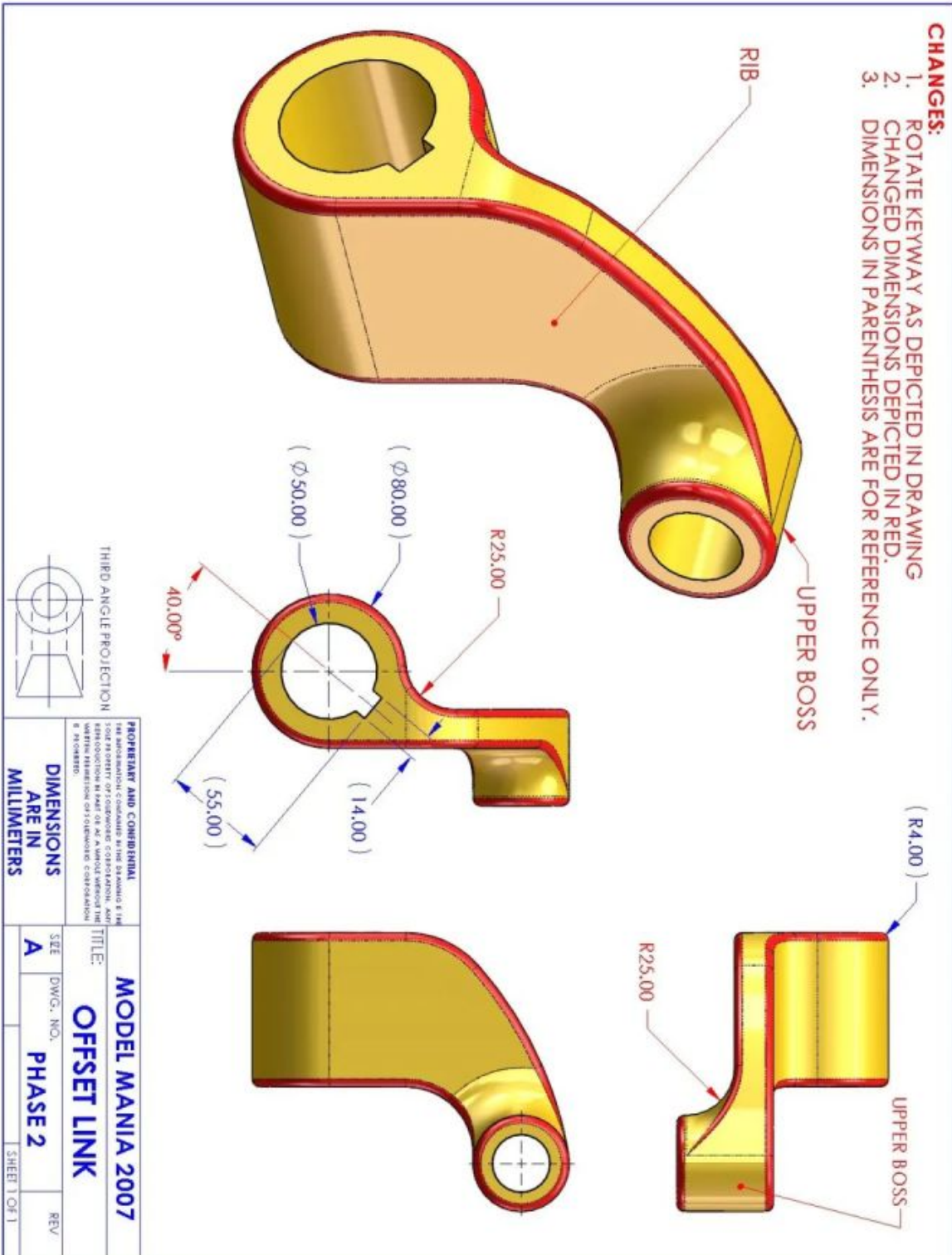






CHANGES:

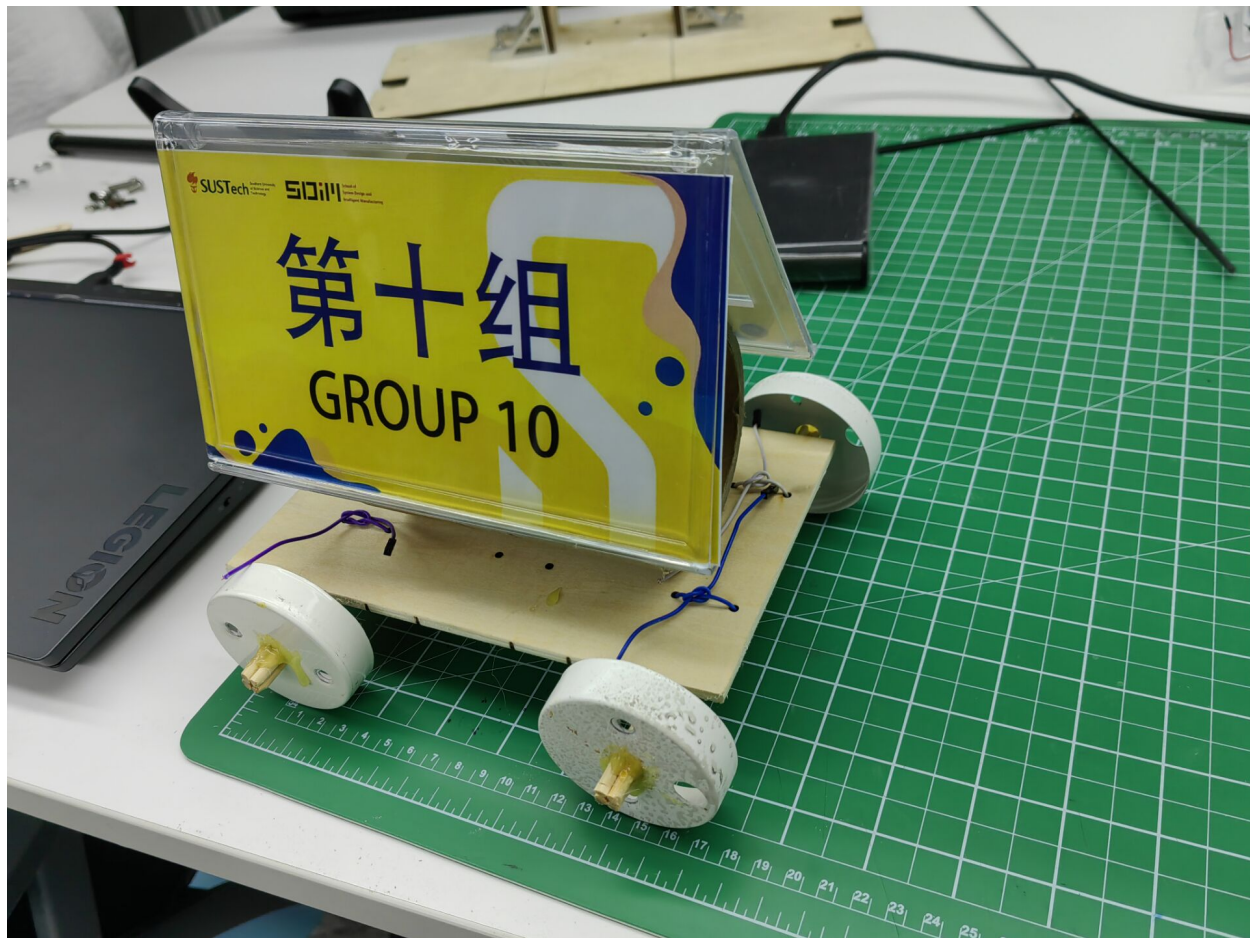
1. ROTATE KEYWAY AS DEPICTED IN DRAWING
2. CHANGED DIMENSIONS DEPICTED IN RED.
3. DIMENSIONS IN PARENTHESIS ARE FOR REFERENCE ONLY.



实际画了两幅图之后，觉得画图还是件蛮有意思的工作

本周工作总结

本周完成了第一代小车的模型，虽然做的很简陋，但是让我们初步对于小车的概念有所理解

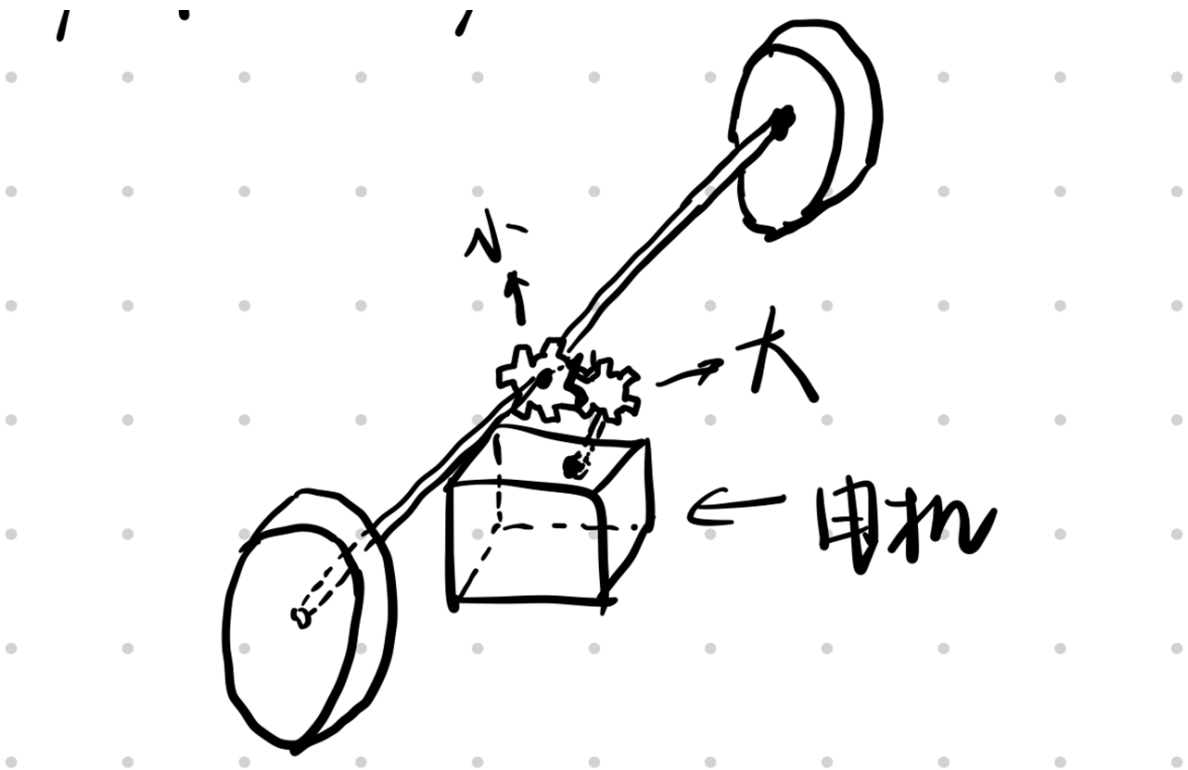
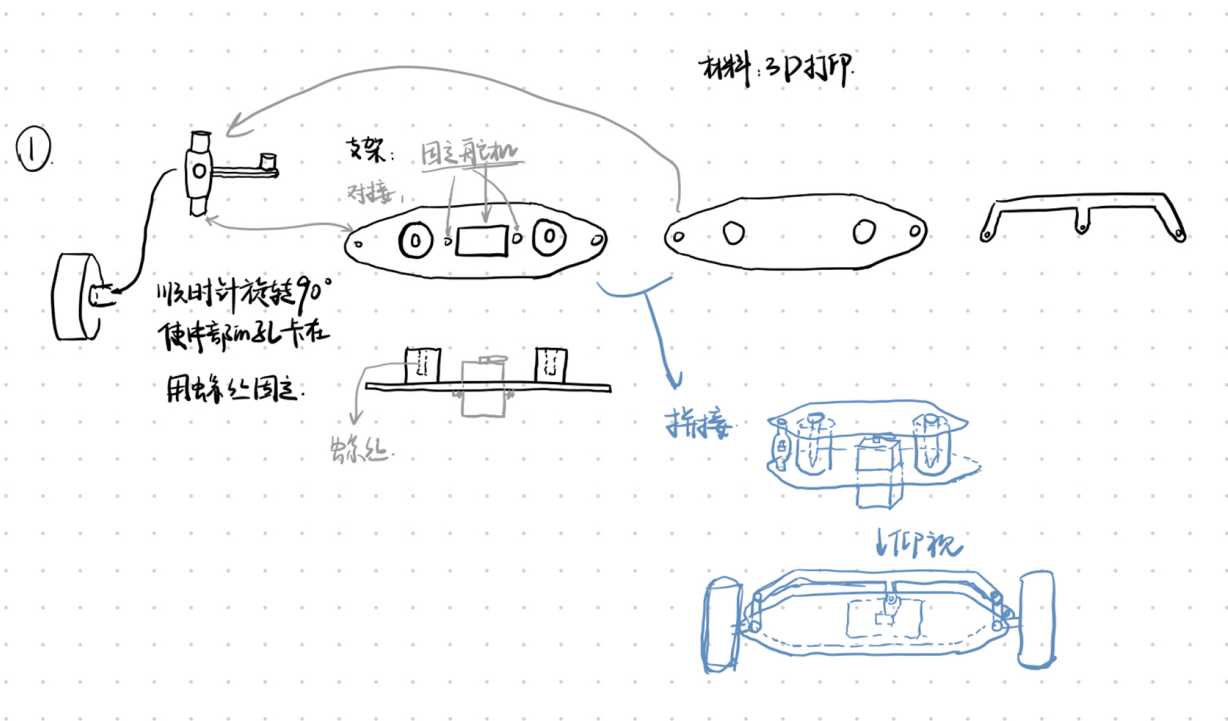


课上3D小车制作展示

我们组根据海绵宝宝的主题设计了这款“菠萝车”，外形如下



关于下周工作内容的讨论



我们将下周需要做的事情初步讨论了一下，预计下周将制作一个能够运动的能够自由转向的带电机的小车，设计图如上，具体数据细节于下周讨论。

