**开源硬件实战总结**

1. **入门**

刚开始接触开源硬件，就了解了一些关于开源硬件的基本常识。这个词主要是用来反映自由释放详细信息的硬件设计，如[电路图](https://baike.so.com/doc/5749007-5961765.html)、材料清单和电路板布局数据，通常使用开源软件来驱动硬件。

共享[逻辑设计](https://baike.so.com/doc/9164612-9497799.html)连同可程式逻辑器件之重构，也是一种形式的开源硬件。[硬件描述语言](https://baike.so.com/doc/9455619-9797274.html)代码的共享代替共享电路图。硬件描述语言通常用於系统晶片系统，也用於元件可程式逻辑闸阵列或直接在[专用集成电路](https://baike.so.com/doc/5466459-5704679.html)中设计。当分配时叫做硬件描述语言模块或 IP cores 。

通过一个概念可以更容易理解开源硬件，那就是"开源软件"，它产生在开源硬件之前，安卓就是开源软件之一。开源硬件和开源软件类似，就是在之前硬件的基础之上进行二次创意。在复制成本上，开源软件的成本也许是零，但是开源硬件不一样，其复制成本较高。

开源硬件延伸着开源软件代码的定义，包括软件、电路原理图、材料清单，设计图等都使用开源许可协议，自由使用分享，完全以开源的方式去授权方式。以往的DIY在分享的时候没有清楚的授权，开源硬件把软件惯用的GPL，CC等协议规范带到硬件分享领域。

开源硬件指与自由及开放原始码软件相同方式设计的计算机和电子硬件。开源硬件开始考虑对软件以外的领域开源，是开源文化的一部分。相对于以Linux为代表的开源[软件](http://www.so.com/s?q=%E8%BD%AF%E4%BB%B6&ie=utf-8&src=internal_wenda_recommend_textn)阵营，[开源硬件](http://www.so.com/s?q=%E5%BC%80%E6%BA%90%E7%A1%AC%E4%BB%B6&ie=utf-8&src=internal_wenda_recommend_textn)也是开 源思想的继承者。这些[硬件](http://www.so.com/s?q=%E7%A1%AC%E4%BB%B6&ie=utf-8&src=internal_wenda_recommend_textn)的[开发者](http://www.so.com/s?q=%E5%BC%80%E5%8F%91%E8%80%85&ie=utf-8&src=internal_wenda_recommend_textn)将硬件的全部资料都对外公开，包括电路图、[固件](http://www.so.com/s?q=%E5%9B%BA%E4%BB%B6&ie=utf-8&src=internal_wenda_recommend_textn)、软件、[元件](http://www.so.com/s?q=%E5%85%83%E4%BB%B6&ie=utf-8&src=internal_wenda_recommend_textn)列表、[器件](http://www.so.com/s?q=%E5%99%A8%E4%BB%B6&ie=utf-8&src=internal_wenda_recommend_textn)列表以及印刷版图。这些资料允许任何人使用，比开 源软件的开放度更高的地方在于，作者更是允许将这些资料及硬件用于任何[商业](http://www.so.com/s?q=%E5%95%86%E4%B8%9A&ie=utf-8&src=internal_wenda_recommend_textn)开发。我们了解了为什么要学习开源硬件，如何学习开源硬件，了解了几个常用软件（arduion，frizting，processing。），常用网站（github，thinkcard。）还了解了一些硬件以及他们的作用。

1. **实践**

在第二天的学习中，用morse代码为例子来为我们展示开源硬件的应用，把26个英文字母转换成morse代码，让我们更深入的了解了c以及其他语言的用处。

#include<stdio.h>  
#include<iostream>  
using namespace std;  
int main()  
{   
char a[26][6]={{'.','-'},{'-','.','.','.'},{'-','.','-','.'},{'-','.','.'}, {'.'},{'.','.','-','.'},{'-','-','.'},{'.','.','.','.'}, {'.','.'},{'.','-','-','-'},{'-','.','-'},{'.','-','.','.'}, {'-','-'},{'-','.'},{'-','-','-'},{'.','-','-','.'}, {'-','-','.','-'},{'.','-','.'},{'.','.','.'},{'-'}, {'.','.','-'},{'.','.','.','-'},{'.','-','-'},{'-','.','.','-'}, {'-','.','-','-'},{'-','-','.','.'}};  
int i,m;  
char str[100];   
char c;   
gets(str);   
printf("%s",str);  
for (i=0;(c=str[i])!='\0';i++)   
if(c==' ') printf(" ");   
else  
printf("%s",a[m=str[i]-97]),printf(" ");  
printf("\n");  
getchar();  
return 0;   
}   
/\*摩尔斯码转英文\*/   
#include<stdio.h>   
#include <string.h>  
#include <iostream>  
using namespace std;  
int main()   
{   
char a[26][6]={{'.','-'},{'-','.','.','.'},{'-','.','-','.'},{'-','.','.'}, {'.'},{'.','.','-','.'},{'-','-','.'},{'.','.','.','.'}, {'.','.'},{'.','-','-','-'},{'-','.','-'},{'.','-','.','.'}, {'-','-'},{'-','.'},{'-','-','-'},{'.','-','-','.'}, {'-','-','.','-'},{'.','-','.'},{'.','.','.'},{'-'}, {'.','.','-'},{'.','.','.','-'},{'.','-','-'},{'-','.','.','-'}, {'-','.','-','-'},{'-','-','.','.'}};  
int b,i,m;  
char str[100],k;   
char c,\*p,t[6];  
gets(str);   
printf("%s",str);   
p=str;  
while(\*p!='\0')   
{   
i=0;   
b=0;  
k=0;  
while(\*p!=' '&&\*p!='\0')  
{  
t[i++]=\*p;   
p++;   
}  
t[i]='\0';  
while(strcmp(a[k++],t)!=0);   
c=k-1+97;   
putchar(c);  
while(\*p==' ')   
{  
b++;p++;}   
if(b==3)   
printf(" ");  
}   
getchar();  
return 0;  
}

1. **深入接触**。

在第三天的学习中，我了解了怎么把程序和硬件联系起来，用程序去实现一些自己的想法，通过软件来模拟虚拟环境从而把程序模拟到硬件中，比如，控制小车的移动，用单片机控制七段数码管显示数字，通过这个实例，我也了解到了一些电路的知识和数码管的基本使用方法。

byte DIGITAL\_DISPLAY[10][8] = { //设置0-9数字所对应数组

{0,1,1,1,1,0,1,1}, // = 0

{0,1,1,0,0,0,0,0}, // = 1

{0,0,1,1,0,1,1,1}, // = 2

{0,1,1,1,0,1,0,1}, // = 3

{0,1,1,0,1,1,0,0}, // = 4

{0,1,0,1,1,1,0,1}, // = 5

{0,1,0,1,1,1,1,1}, // = 6

{0,1,1,1,0,0,0,0}, // = 7

{0,1,1,1,1,1,1,1}, // = 8

{0,1,1,1,1,1,0,1} // = 9

};

void setup()

{ //设定4-11号数字端口为输出

Serial.begin(9600);

for(int i=4;i<=11;i++){

pinMode(i, OUTPUT);

}

}

int income;

void loop()

{

//0-9数字显示

if(Serial.available()>0){

income = Serial.read();

}

int pin = 4;

switch(income)

{

case '0':

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

{

digitalWrite(pin, DIGITAL\_DISPLAY[0][s]);

pin++;

}

break;

case '1':

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

{

digitalWrite(pin, DIGITAL\_DISPLAY[1][s]);

pin++;

}

break;

case '2':

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

{

digitalWrite(pin, DIGITAL\_DISPLAY[2][s]);

pin++;

}

break;

case '3':

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

{

digitalWrite(pin, DIGITAL\_DISPLAY[3][s]);

pin++;

}

break;

case '4':

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

{

digitalWrite(pin, DIGITAL\_DISPLAY[4][s]);

pin++;

}

break;

case '5':

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

{

digitalWrite(pin, DIGITAL\_DISPLAY[5][s]);

pin++;

}

break;

case '6':

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

{

digitalWrite(pin, DIGITAL\_DISPLAY[6][s]);

pin++;

}

case '7':

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

{

digitalWrite(pin, DIGITAL\_DISPLAY[7][s]);

pin++;

}

case '8':

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

{

digitalWrite(pin, DIGITAL\_DISPLAY[8][s]);

pin++;

}

case '9':

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

{

digitalWrite(pin, DIGITAL\_DISPLAY[9][s]);

pin++;

}

}

delay(1000);

}

1. **总结。**

开源硬件并不仅仅是硬件设计方法的开放，而更多地是体现了一种创新理念的开放。这种理念坚信从分享中所获多于自身付出。当开发者不再受专利授权所困，越来越多地公开分享他们的创新时，他们便能借此获得越来越多的免费帮助，进而改进自己的发明。我们不断创造的同时也在不断改进所使用的开源硬件，使之不断完善。 而这个过程中恰恰也是一个学习者成长的过程，学习者个体从这个过程中收获的，是包括人际沟通、团队协作、创新问题解决、批判性思维和专业技能等在内的全方位的成长。