基于可见光通信的移动存储设备：light-U
Mobile storage device based on visible light communication

519021910528 郑心浩，519021910548 刘雨田, 519021910366 窦铱明
void_zxh@sjtu.edu.cn，stau7001@sjtu.edu.cn, douyiming@sjtu.edu.cn
项目主页：https://github.com/Dou-Yiming/CS339-Project

2021 年 12 月 13 日

目录

1 简介与意义

1.1 项目简介

本项目受可见光通信技术（VLC，Visible Light Communication）的启发，提出并实现了一种基于 VLC 的移动存储设备——light-U。

light-U 可通过可见光发生器的高频闪烁传输二进制编码的信号，接收端对信号进行滤波后以协议规定的原则解码即可得到文件。

首先，我们设计并实现了一种信号编码机制将任意格式的文件编码为二进制的、可以发送的数据包，进而设计传输协议在数据段首部和尾部分别加入起始信息以及错误校验码，实现了数据信号的稳定、容错的数据传输。

另外，我们设计并使用 arduino 套件搭建了全套的可见光发生器、接收器以及存储器，通过将软件协议写入硬件，实现了可见光信息的发送与接收。

在完成了软硬件设计后，我们进一步通过 solidworks 建模等方式设计了 light-U 的外壳模型，将传输设备安装于其中后将 light-U 实现为具有实用价值且美观的移动存储设备。

简言之，这一项目的创新性贡献如下：

- 设计并实现了一套全新的可见光传输协议用以编码、解码可见光信号
- 设计并搭建了发射、接收、存储可见光信号的全套电路硬件
- 设计并通过 3D 打印的方式将 light-U 制作为较为稳定、易用、美观的“产品级”移动存储设备

1.2 项目框架

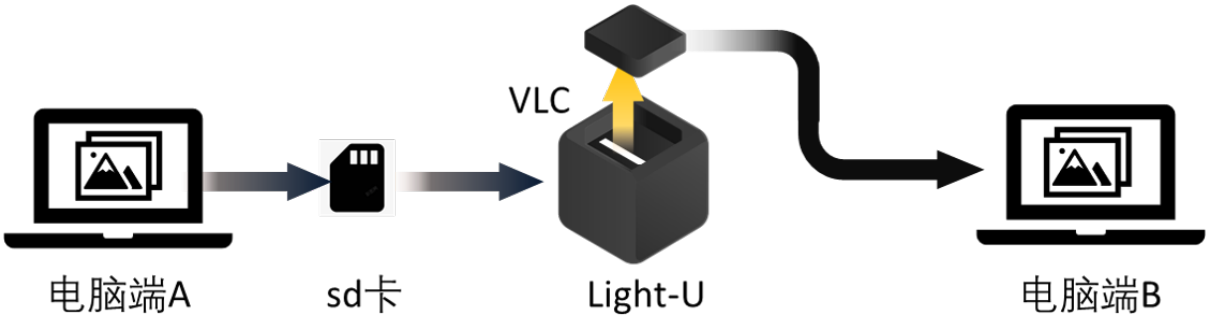

图 1: Light-U 项目框架

在实践过程中，本项目的方法框架如上图所示。与传统信号传输结构类似，本项目从发送端电脑出发，将数据传输至移动储存端的 Light-U 储存模块上，并经由移动端发送模块，在密闭的光环境下，将数据由可见光的形式传输至接收端的接收模块，并在接收端电脑处对数据进行解包，从而实现类似移动硬盘的效果。

具体而言，我们先将发送端电脑的数据包经由串口拷贝到移动端 Light-U 的储存设备中，并对数据进行进一步地编码拆分。接着，我们通过发送模块，利用可见光调制技术，将数据通过可见光的形式发送至接收端接收模块中。在接收模块中，我们对数据进行解调，并初步整合数据，将其通过串口传递至接收端电脑中。最终在接收端电脑中对数据进行类型识别、整合等最终处理，从而实现整个文件的传递过程。

2 相关工作

可见光通信 可见光通信是一种利用波长在 380 nm 到 790nm 范围内的可见光进行数据通信的无线光传输技术 [1]，作为一种低功耗、高带宽、低电磁干扰的新兴传输技术，可见光通信具有较大范围的频谱带宽（400THz），且位于传统无线通信未使用的频谱段（400 到 800THz），因此可以有效填补日益增长的频谱需求。除此之外，在高电磁干扰的应用场景，可见光可以替代其他无线通信技术完成正常传输。其较强的指向性与快速的衰减速率也使得可见光通信的可以通过多输入多输出技术（MIMO，Multiple Input Multiple Output）提高空间复用率。同时，由于可见光兼具照明功能，使用可见光通信能在一定程度上减少能源消耗，达到节能减排的效果。

目前的可见光通信通常使用调制带宽高的白光 LED（发光二极管）作为发光光源，PD（Photodiode）作为接收端。本项目最终实现也采用了 LED 作为发送端的硬件实现。LED 用于可见光通信，相比其他光源具有以下几种优势：

- 传输所需的能耗较低。LED 的发光效率约为 249 lm/W，比普通日光灯高了 3 倍。
- 平均发光寿命长，光衰小。普通 LED 的平均寿命高达 10^5 h，一年光衰不到 3%，可以为 VLC 提供可靠的长期硬件支持。
- 易于安装和更换。LED 不需要额外的辅助配置，如整流器等，大大降低了安装和日常维护的难度。

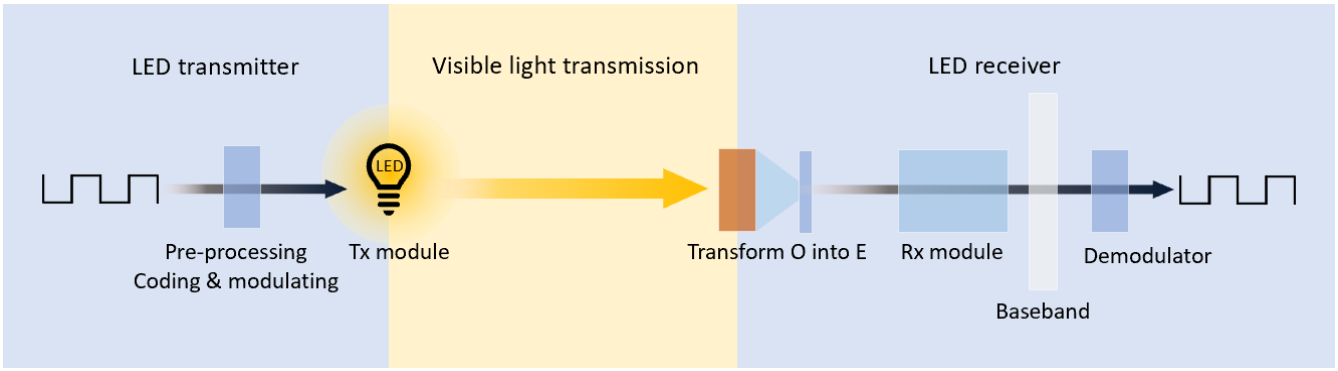

图 2: 可见光通信过程

一般的可见光通信包含以下三个基本过程：信号发射、信道传输和信号接收，即通过对二进制比特流进行编码调制后输出到 LED，从而将电信号转化为光信号。然而在实际应用中，LED 的调制带宽往往成为了限制可见光通信的速率的主要因素。因此，许多相关的研究工作都从不同的方面提出了优化这一问题的方法。比较有代表性的工作包括：正交频分复用、无载波幅相调制等针对调制技术的优化技术、双级联幅度均衡器 [2]、FTR 滤波器 [3] 等硬件与软件预均衡技术、以混合后均衡为代表的后均衡技术以及针对 LED 光学材料的优化。

可见光调制 最早被提出的可见光调制技术为开关键控技术（OOK，On-Off Keying），脉冲位置调制技术（PPM，Pulse Position Modulation）紧随其后。作为 PPM 的改进，多脉冲位置调制（MPPM，Multiple Pulse Position Modulation）[4] 由于其较高的带宽利用率而得到了广泛的研究和应用。

随着多输入多输出技术（MIMO，Multiple Input Multiple Output）在可见光通信中的发展，被用于解决 MIMO 信号相关性的调制技术——空间调制技术（SM，Spatial Modulation）[5] 应运而生，即在发射时只激活单根天线进行传输。其在可见光通信领域的发展包括光空间调制技术（OSM，Optical Spatial Modulation）[6]，使用多个 LED 的广义空间调制技术（GSM，Generalized Spatial Modulation）[7] 等等。

另一方面，考虑到可见光通信的照明功能，控制传输过程中的灯光强度也成为了一个较为热门的研究方向。基于不同调制系统的 PWM 调光相继被提出，进一步拓展了可见光通信的应用场景。

3 数据传输编码及协议设计

在本项目中，我们将数据信号传输模块分为数据编码以及协议设计两个部分。首先需要将待传输的数据转换为 0、1 表示的二进制编码，进而设计传输协议将数据包以可见光闪烁的形式进行发送。

3.1 数据段编码方式

可见光传输过程中，通过控制电压强弱，可以使灯泡发出亮暗两种状态。我们可以将灯光的亮与暗分别表示为 1、0 两种状态，进而发送二进制编码。

因此，不论传输何种数据，都需要先将数据统一编码为二进制串的格式。在本实验中，我们选用了单通道图片进行传输，这样选取的理由如下：

1. 单通道图片可以视作二维矩阵，每一像素均由一个 $0-255$ 的值表示，因此可以使用统一、简单的 8 位二进制格式编码每一像素
2. 图片中的每一像素均是独立的数据，如果传输过程中出现细微差错（比如将某一编码中的一位记错）不会剧烈影响整体性，保证解码过程可以成功执行、得到结果
3. 图片大小可以随意改变，使得我们可以以较简便的方式完成对比实验

编码过程中，我们将图片中的每一像素均使用 8 位二进制表示，因此对于一张 $N \times N$ 的单通道图片，我们可以使用大小为 $N \times N \times 8 = 8N^2$ 的二进制数组进行编码。图片编码过程如下图所示：

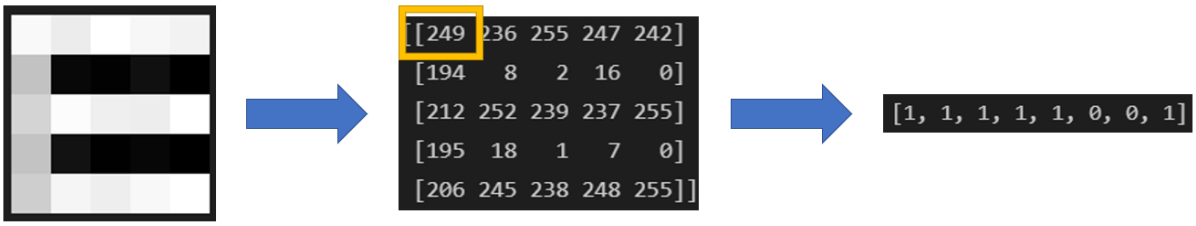

图 3: 图片编码过程

发送过程中，将按序遍历整个数组并发送每一像素的二进制数据。

3.2 数据帧结构

将数据成功编码后，还需要设计数据帧将数据以“包”的方式进行发送。数据帧的主要部分及其作用如下：

1. 帧头：标志着数据传输的开始，用以“提示”接收方开始进行信息接收，使得接收方和发送方的信息同步起来。
2. 数据部分：数据帧的主要部分，包含着二进制编码数据
3. 帧尾：在数据传输过程中，1 和 0 可能会发生互换，产生比特差错。数据帧帧尾主要用于校验数据发送是否出现差错

本实验中，每一数据帧中包含 1 字节由于只涉及两台机器间的通信过程，故无需设计首部的目标地址字段，因此简单采用 4 位 1 作为首部。当接收方收到连续的 4 位 1 后，便开始记录后续的一字节作为数据。在数据帧末尾，我们使用冗余码进行校验。我们使用的数据帧结构如下所示：

图 4: 数据帧结构

4 实验电路模块搭建

在本项目中，我们将分为发送模块、接收模块、储存模块三部分搭建可见光传输系统，实现由发送端电脑中读取数据，存入存储模块中。在将 light-U 与接收端电脑连接，经由发送模块发送可见光信号，由接收模块读取信号后，将数据转移至接受端电脑中，实现光 U 盘的可见光传输功能。

4.1 发送模块

在本项目中，发送模块要具有独立供电、按键触发、发送光信号等基础功能。基于硬件条件以及经费限制，我们设计搭建以下光信号发送模块电路。

- **硬件设备**：Ardinuo Nano 开发板、按键开关、9V 干电池、发光二极管、杜邦线若干。
- **工作流程**：

  依据 Ardinuo Nano 开发板供电需求，我们采用 9V 的干电池作为外部电源进行供电，利用按键开关来控制信号的发送，使用发光二极管作为发光媒介来传递光信号。在实验阶段，我们仅对光进行幅度与频率的调制，记高亮度、规律性高频闪为 1，低亮度、规律性低频闪为 0。其具体工作流程如下：

  1. 初始化接口输出，发光二极管电压正极电压置 0，为按键开关输入端配置上拉电阻。
  2. 读取按键开关接口数据，若为高电平，进入 step 3; 反之，重复 step 2。
  3. 按照信号通讯协议，将需发送的数据包从存储模块中取出，将数据包分段，并将每个数据段分解为二进制位并调制后，经由发光二极管输出光信号。
  4. 输出完成后，清空 Ardinuo Nano 开发板内存数据，返回 step 2 等待下一次信号传输。

- **模块电路图**：

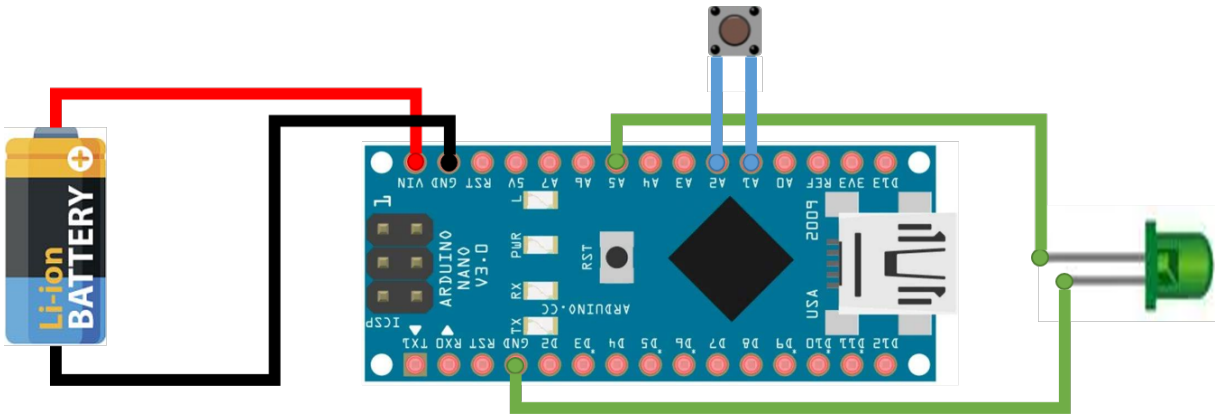

图 5: 发送模块电路

- **核心实现代码**：

```
//获取启动信息
int button = digitalRead(A2);
if(button==0)
{
    for(int i=0;i<n;++i)
        for(int j=0;j<n;++j)
        {
            //数据帧首部输出
            digitalWrite(Outpin,HIGH);
            delay(20);
            //单字节信号输出
            for (int k=0;k<8;++k)
            {
                int out=a[i][j][k];
                if(out==0)
                {
                    digitalWrite(Outpin,LOW);
                    delay(5);
                }
                else if (out==1)
                {
                    digitalWrite(Outpin,HIGH);
                    delay(5);
                }
            }
        }
        //校验
        if(last==0)
        {
            digitalWrite(Outpin,HIGH);
            delay(10);
        }
        else
        {
            digitalWrite(Outpin,LOW);
            delay(10);
        }
}
```

4.2 接收模块

在本项目中，接收模块要具有接收光信号、波形整形、与接收端电脑进行串口通信等基础功能。基于此，我们搭建以下设计搭建接受模块。

- **硬件设备**：Ardinuo Nano 开发板、光敏电阻、定值电阻、MiniUSB 串口通信线、接收端电脑、杜邦线若干。
- **工作流程**：

  在接受模块的电路中，由于接受模块作为接收端电脑的一个借口存在，我们采用 MiniUSB 串口通信线直连的方式对 Ardinuo Nano 开发板进行供电。

在接收模块中，我们利用光敏电阻感应连续的光信号变化，通过对端口的电压进行采样，从而得到离散的电压值，依据设计的电路，在光亮度较大时，采样得到低电压；在光亮度较小时，采样得到高电压。该模块具体工作流程如下：

1. 初始化接口输出，与接收端电脑建立串口通信
2. 利用光敏电阻实时感应连续的光信号数据，对光敏电阻与定值电阻连接端端口电压进行采样记录
3. 将采样的到的离散电压波形进行整形，得到 01 形式的数据串
4. 判定采样得到的信号中是否出现数据包首部，若出现，则进入 step 5; 反之继续采样。
5. 读取数据包中数据，直至出现帧尾空白数据，将数据经由串口输出至接收端电脑后，返回 step 4 继续采样
6. 在接受端依据数据类型，对数据包进行进一步的整合，得到具体类型的最终数据，完成传输

- **模块电路图**：

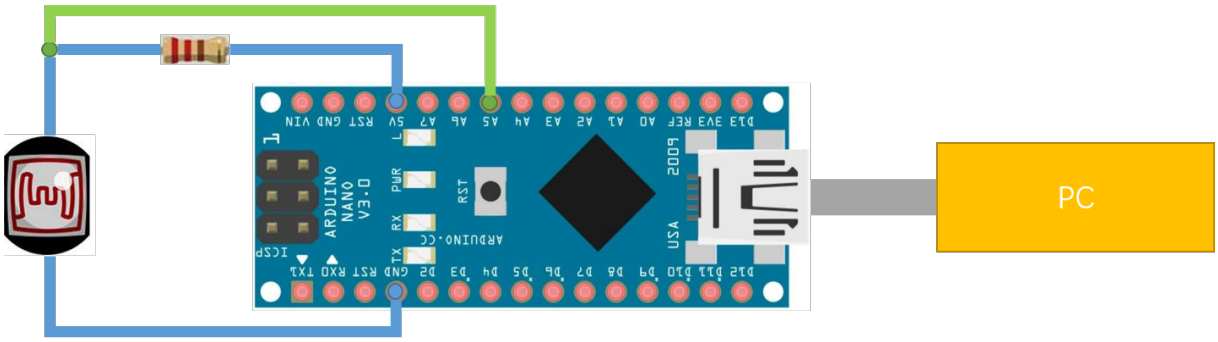

图 6: 接收模块电路

- **核心实现代码**：

光信号采集与波形初步整形（Ardinuo Nano）

```
Intensity = analogRead(AD5);  //读取模拟口AD5的值，存入Intensity变量
cur_time = millis();
cur_state=get_state(Intensity);
if(cur_state!=state) // state change
{
    hold_time=cur_time - pre_time;
    int cnt=(int)((double)hold_time/(double)5+0.5);
    next:
    if(cnt>=4 && state==1 && start==false && pos==0) // 开始记录
    {
        start=true;
        cnt-=4;
        Serial.println("start");
    }
    else if (cnt>=4 && state==1 && start==true) //上一轮还没记录完
    {
        int tmp=8-pos;
        cnt-=tmp;
        for(int i=0;i<tmp;++i)
            ans[pos++]=state;
```

```
        if(pos==8)//一个包记录完成
        {
            for(int i=0;i<8;++i)
            {
                Serial.println(ans[i]);
                id++;
            }
        }
        pos=0;
        start=false;
        state=1;
        goto next;
    }
    if(start==false)
        Serial.println("continue");
    else
    {
        for(int i=0;i<cnt;++i)
            ans[pos++]=state;
        if(pos==8)//一个包记录完成
        {
            pos=0;
            for(int i=0;i<8;++i)
            {
                Serial.println(ans[i]);
                id++;
            }
            start=false;
        }
    }
    pre_time = millis();
    state=cur_state;
  }
```

数据处理（接受端电脑 python 脚本）

– 串口类定义与初始化

```
class SerialCommunication:
    def __init__(self, port='com3', baud_rate=9600, timeout=0.0001):
        self.port = port
        self.baud_rate = baud_rate
        self.timeout = timeout
        self.serial = serial.Serial(
            port=self.port, baudrate=self.baud_rate, timeout=timeout
               )

    def openSerial(self):
        if self.serial.isOpen():
            print('串口已经打开!')
        else:
            self.serial.open()
            print('串口打开成功!')

```

```
    def printInfo(self):
        print('name: ', self.serial.name)
        print('port: ', self.serial.port)
        print('baud_rate: ', self.serial.baudrate)
        print('timeout: ', self.serial.timeout)

    def closeSerial(self):
        self.serial.close()
        print('串口已关闭! ')
```

– 数据获取子模块

```
def get_data():
    buffer = serial.serial.readline()
    data = str(buffer[:-2])[2:-1]
    if len(data) > 0:
        return int(data)
    return


def loop():
    global step, intensity_list, state_list, bit_list
    step += 1
    if step < 10:
        return
    data_bit = get_data()
    if not data_bit == None:
        bit_list.append(data_bit)
    if len(bit_list) == 8:
        value = 0
        for i in range(8):
            value += bit_list[7-i]*pow(2, i)
        img.append(value)
        bit_list = []
    if len(img) == 25:
        return True
    return
```

– 数据处理主程序

```
if __name__ == '__main__':
    # 读取串口数据
    serial = SerialCommunication(port='com5', baud_rate=115200,
       timeout=0.5)
    serial.openSerial()
    while True:
        if loop():
            break
    serial.closeSerial()
    serial.printInfo()
    # 整合形成图片输出
    img=np.array(img).reshape((5,5))
    print(img.shape)
    img = Image.fromarray(img.astype('uint8'))
```

```
14      img.save("Pic1.jpg")
```

4.3 储存模块

在本项目中，由于开发板本身的存储空间十分有限，我们需要引入额外的存储模块来存储我们要发送或是接收到的数据。储存模块要具有独立存储、以及实时读取等功能，并且作为附加模块用于发送模块或接受模块（储存模块与其共用同一开发板）。基于此，我们搭建以下设计搭建储存模块。

- **硬件设备**：Ardinuo Nano 开发板、Micro SDK 卡模块、9V 干电池、杜邦线若干
- **工作流程**：

  在储存模块中，Ardinuo Nano 开发板的供电方式由储存电路连接的模块决定，我们使用 Micro SDK 卡模块作为电路额外的存储，通过 SDK 模块的 MOSI 引脚写入 SDK，通过 MISO 引脚读取 SDK 中的文件。该模块功能工作流程如下：

  1. 初始化接口输出与 SD 卡，开启 SD 卡模块的串口连接。
  2. 根据需要进行以下操作

     读取文件：若需要读取文件，利用 Ardinuo 中 `SD.h` 库调用 `open()` 函数，开启 SD 卡中的文件，利用 `read()` 函数读取文件中的数据。

     写入文件：若需要写入文件，同样利用 Ardinuo 中 `SD.h` 库调用 `open()` 函数，开启 SD 卡中的文件，利用 `print()` 函数写入文件。

- **模块电路图**：

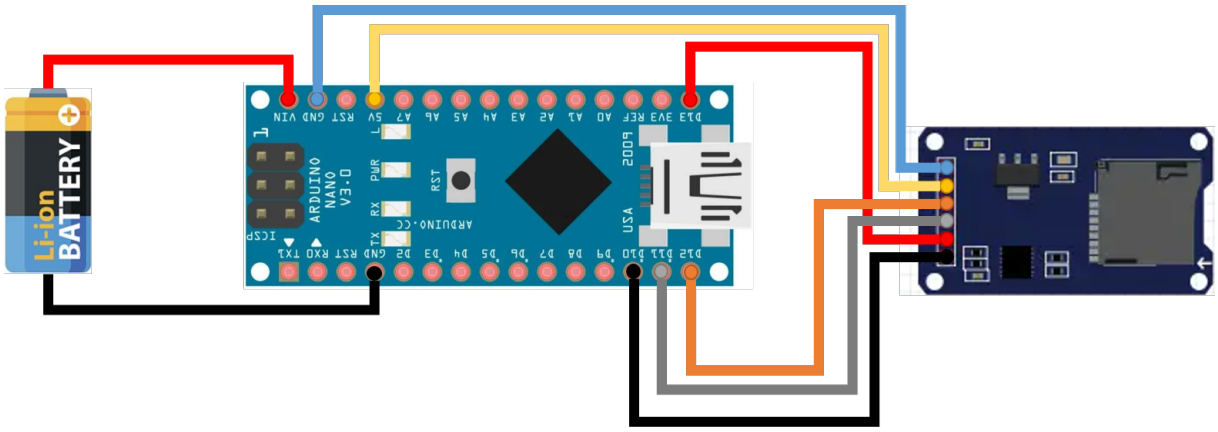

图 7: 储存模块电路

- **核心实现代码**：

  初始化 SD 卡模块：

```
1  #include <SPI.h>
2  #include <SD.h>
3
4  File myFile;
5  void setup()
6  {
7      Serial.begin(9600);
8      if (!SD.begin())
9      {
```

```
        Serial.println("Open failed");
        return;
    }
    Serial.println("Open successfully.");
}
```

读取文件:

```
datafile = SD.open("image.txt", FILE_WRITE);
if (myFile)
{
    //写入数据
    for(int i=0;i<n;i++)
        myFile.println(data[i]);
    myFile.close();
    //分隔符
    Serial.println("-----------------------");
}
else
    Serial.println("Error occur when opening the file");
```

写入文件:

```
myFile = SD.open("test.txt");
if (myFile)
{
    //读取有限常的部分数据
    cnt=0;
    while (myFile.available()&&cnt<limit_len)
    {
        Serial.write(myFile.read());
        cnt++;
    }
    myFile.close();
}
else
    Serial.println("Error occur when opening the file");
```

5 信号整形方案

Arduino 中读出的数据为光敏电阻感受到的光强，且由于灯泡点亮以及熄灭均需要一定时间，信号中存在着一定噪声干扰，使之并非理想的方波。因此我们需要对这一信号进行滤波操作后才能得到二进制数据。在本项目中，我们主要采用基于阈值进行滤波以及利用快速傅里叶变换(FFT)进行滤波。

5.1 阈值滤波

阈值滤波的基本思想是，在灯光点亮或者熄灭时，其亮度均会高于或者低于某一阈值。我们可以通过人为设定这一阈值的方式将信号整形为方波，具体过程如下图所示:

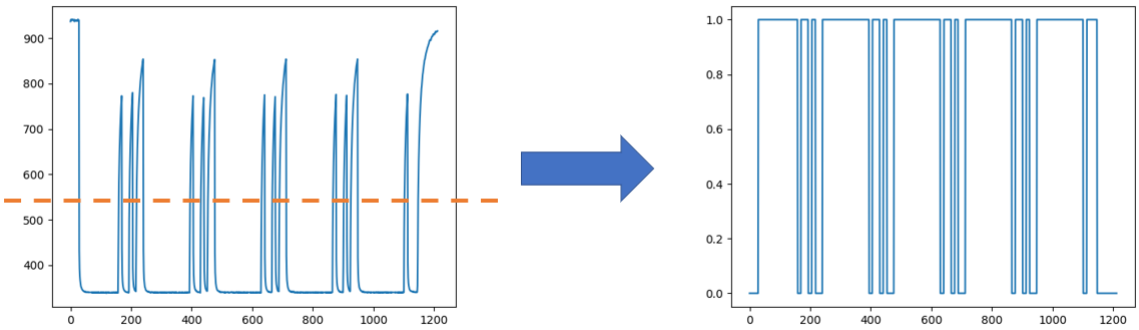

图 8: 阈值滤波

可以看出通过设定大小适中的阈值，我们可以将信号中高于或低于这一阈值的部分分别置为 0 或者 1，进而将信号整形为非常整齐的方波波形，为后续的信息处理做好准备。

5.2 频域滤波

根据傅里叶变换的思想，任何一种时域上的函数都可以分解为多个正弦函数地线性组合，也即频域上的函数。因此在频域滤波技术中，我们主要采用了快速傅里叶变换的方式将得到的时域信号离散化地表示为频域信号，进而可以将信号中的低频噪声进行滤波，从而保留信号中的有效信息。具体流程如下所示：

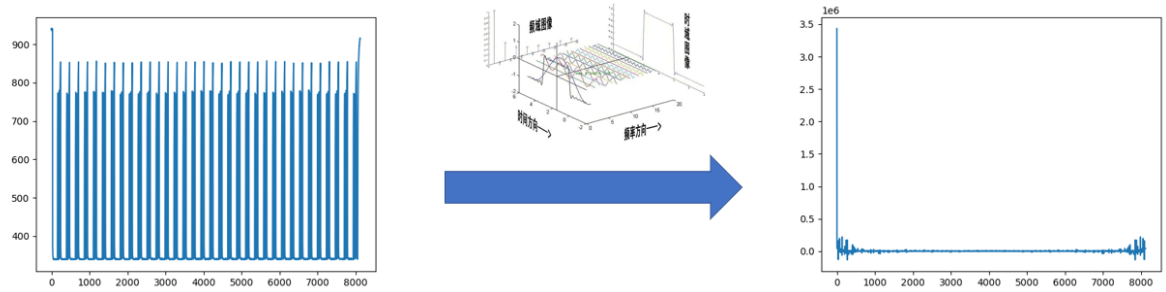

图 9: 通过快速傅里叶变换将时域信号转换为频域信号

通过在频域上选取相应的区间，可以达到在时域上进行滤波的效果。

5.3 基于深度学习的滤波

上述两种滤波法主要基于人工设计的滤波方案，尽管操作简单、较为高效，但对于各种噪声以及异常值的鲁棒性不强，很容易受到干扰。尤其本项目中的信号是使用 LED 灯产生的，其光强并非在电路导通的一瞬间就提升至峰值，也并非在电路断开的一瞬间就彻底熄灭，这一过程持续时间大约为 1ms。

这将会导致我们无法使 LED 灯以高于 1000Hz 的频率闪烁，也因此限制了这一可见光通信系统的极限传输速率。

因此，我们思考可以借鉴机器翻译的方法，将有着明显噪声、传统方法无法识别的的高频时序信号作为 transformer 神经网络的输入，通过提取其中特征的方式解码出正确方波，进而提升系统的极限传输速率。这一方法的构想如下图所示：

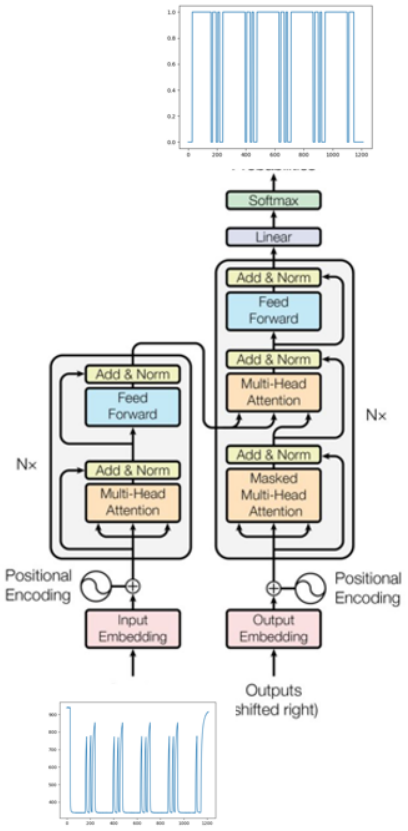

图 10: 利用深度学习进行信号滤波

基于此，我们可以实现更为有效且鲁棒的信号整形效果。

在实验中，由于第三种方式需要大量采集原始电压数据进行训练并在应用时调用模型进行处理，所需时间较长，为了提高信号整形速度，我们采用前两种方法为主，进行信号的整形。

6 外观设计

为了减少传输过程中其他可见光的干扰、保护电路模块，本项目设计并建模了一组用于装载通信电路的分体式外壳。

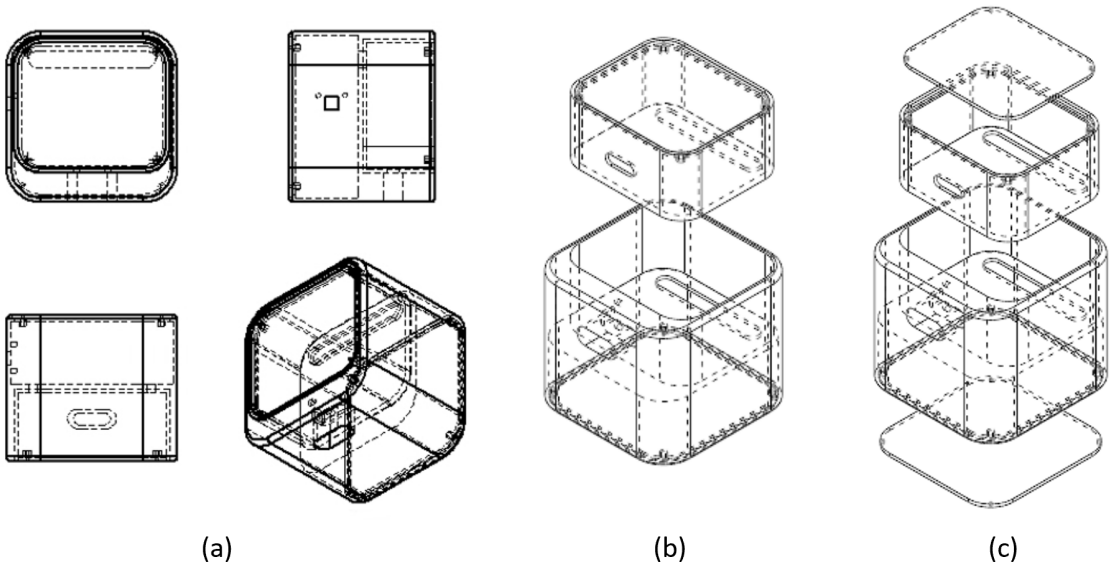

图 11: 分体式外壳

如上图所示，light-U 由移动存储端和接收端两部分构成，读取数据或一起存放时，接收端可放入移动端的上层空隙中，如图（a）所示（下文简称这一状态为合体态）。

接收端的 Adruino nano 主板正面朝下插于小型面包板上，mini-B USB 插口固定于侧面的定位槽口。接受端的下壳有一条槽口状的镂空，安装有亚克力透明挡板，光敏电阻被固定于槽口的轴线下方。

移动存储端的 Adruino nano 主板正面朝上，插在连有 sd 卡、9V 电源和 LED 的小型面包板上。移动存储端的矩形孔隙的上壳也有一个与接受端形状相同的槽口镂空，安装有透明亚克力板，LED 被固定于槽口的轴线下方。当 light-U 处于合体态时，接收端和发送端的槽口将完全贴合。此时光敏电阻将与 LED 位于同一封闭空间中并且空间距离极近。该空间中不存在其他干扰光源，从而避免了环境光的干扰，减少了传输过程中外界噪声的干扰。

读取数据时，首先将接收端亚克力板朝下放于移动端的正确位置，将数据线插入接收端的 mini-B USB 口，确认电脑与接收端主板连接良好后，就可以启动移动存储端，开始进行数据传输。

考虑到本项目的高保密性移动存储定位，故接收端体积较小，便于转移和藏匿。使用时可以单独携带移动存储端，以保护存储数据的安全。

规格尺寸： 合体态的 light-U 大小约为 $70 \times 70 \times 58.5mm$，有足够的空间容纳两个 Adruino nano 主板、一个 9V 电池组、两块小型面包板、sd 卡读取模块以及若干电阻、LED 元件。壁厚 1.5mm，材料选择韧性较好的尼龙。壳盖组装方式采用热熔滚花螺母工艺，螺丝规格统一为 M1.2。

具体尺寸与模型细节可查看随附在附件中的工程文件。

7 总结与展望

在本项目中，我们提出、设计并制作了一款基于可见光通信的移动存储设备：light-U。

其功能与传统的 U 盘类似，可以将储存于其中的数据写入另一设备。与传统 U 盘不同的是，light-U 连接的两台设备之间不需要有任何的物理线缆介质，而只需要可见光，相较于传统方法可带来如下几点提升：

1. **保密性**：可见光传输只使用了短距离的光作为介质，传输过程中其信息无法被任何设备捕获，因此具有高度的保密性，可用以传输涉及机密的文件；
2. **并行性**：可见光传输的接收端可以有多台设备同时接收可见光，且传输过程中可以使用不同波长的可见光同时传输不同的数据，从而达到相当高的并行度；

3. **安全性**：light-U 被设计为只有输出端口而无没有写入端口，因此其中的文件为只读文件，因此从根本上杜绝了其中数据被篡改的可能

我们对可见光通信的前景感到非常乐观，同时对基于可见光通信的移动存储设备有着以下几方面的展望与期待：

1. **更先进的滤波技术**：本项目中主要使用阈值滤波、频域滤波等传统技术进行信号的整形，其精度有限。我们期待通过深度学习等先进技术进一步信号滤波、解码过程的效率以及准确性，从而进一步提升信号频率；

2. **更高性能的信号发生器**：本项目中仅仅使用 LED 灯珠作为可见光信号发生器，我们期待将可见光发生器升级为灵敏度更高的设备，从而产生频率更高、波形更加稳定的信号，提升传输效率；

3. **更规范、容错率更高的协议**：本项目中的协议设计对于异常情况的应对能力有限，在复杂环境中易发生错误。我们展望从计算机网络的各种协议中获得灵感，设计出更加规范、在复杂情形中表现更加稳定的传输协议

8 人员与分工

本项目中，我们将需要完成的任务分为传输协议设计、硬件结构设计以及外观设计三个模块，每一模块都对项目的完成极为重要。我们小组分工明确，各自负责其中一个模块并且与其他模块之间相互协调完成这一项目：

- 郑心浩：主要负责项目电路结构的设计以及硬件布局、搭建
- 刘雨田：主要负责 light-U 的外形结构设计、3D 打印以及效果图渲染
- 窦铱明：主要负责传输协议设计以及核心代码的实现与调试

参考文献

[1] Chi N. LED Visible Light Communication Technologies[M]. Beijing: Tsinghua University Press, 2013. 迟楠. LED 可见光通信技术 [M]. 北京: 清华大学出版社, 2013.

[2] H黄星星、陈思源、王智鑫、王一光、迟楠. "1.2 Gbit/s Visible Light Transmission Based on Orthogonal Frequency-division Multiplexing Using a Phosphorescent White Light-emitting Diode and a Pre-equalization Circuit." Chinese Optics Letters 13.10 (2015): 26-29. Web.

[3] Wang, Yiguang, Li Tao, Yuanquan Wang, and Nan Chi. "High Speed WDM VLC System Based on Multi-Band CAP64 With Weighted Pre-Equalization and Modified CMMA Based Post-Equalization." IEEE Communications Letters 18.10 (2014): 1719-722. Web.

[4] Singh, P. K, and Y. K Chauhan. "Performance Analysis of Multi-pulse Electronic Load Controllers for Self-excited Induction Generator." 2013 International Conference on Energy Efficient Technologies for Sustainability (2013): 1299-307. Web.

[5] Mesleh, R.Y, H. Haas, S. Sinanovic, Chang Wook Ahn, and Sangboh Yun. "Spatial Modulation." IEEE Transactions on Vehicular Technology 57.4 (2008): 2228-241. Web.

[6] Mesleh, Raed, Hany Elgala, and Harald Haas. "Optical Spatial Modulation." Journal of Optical Communications and Networking 3.3 (2011): 234-44. Web.

[7] Jeganathan, J., A. Ghrayeb, and L. Szczecinski. "Generalized Space Shift Keying Modulation for MIMO Channels." 2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (2008): 1-5. Web.