整个项目的流程：

整个流程主要包括：

1. **结合不同传感器性能，选择传感器进行方案的设定。**

**（1）项目开始的时候，先使用mpu9250进行方案的设定。**首先考虑到手掌骨架的三维框架模型，可以使用距离或者角度计算对手指关节进行控制。

**首先方案使用加速度传感器数据对手掌模型进行控制。**使用arduino读取一个mpu9250传感器的加速度数据（mpu9250\_getMessage.ino），然后在matlab（inertial\_pdr.m）上面对采集到的数据进行可行性分析。使用ZUPT+EKF模型对加速度数据进行校准并且二次积分。通过多次试验发现，在传感器静止的情况下，积分出来的距离也是很大的；并且在移动相对距离的情况下，积分出来的距离和实际的距离之间差距比较大。在手掌模型中，需要非常精确的距离数据，才能很好地对手指间关节进行比较精确的控制，达到控制的效果。

**其次选择陀螺仪传感器对手掌模型进行控制（mpu9250\_getMessage.ino文件读取传感器数据）。**其中使用zupt+ekf模型对角度数据进行校准，然后控制一个环状物体进行旋转（inertial\_pdr.m和getPic.m文件），通过改变卡尔曼滤波器状态误差和zupt阈值，得到的物体旋转效果还是不好。静止状态的情况下，物体仍然有偏移的状况出现。这种方案在可行性方面也被否定。

（不代表不能做，只是对于这种陀螺仪数据上的偏移问题，短时间解决有点困难）。

**（2）其次，考虑到使用jy901传感器进行直接旋转角度的读取。**

通过查找jy901传感器的相关文档，得知jy901芯片本身能够直接输出roll yaw pitch三个角度，并且三个角度的获得使用了卡尔曼滤波器。下面调用（jy901.ino）读取传感器数据，也可以使用（ReadComData.m）直接读取串口数据，实时显示在MATLAB上面控制立方体的运动。通过实验验证，jy901的三个角度的输出角度漂移比较小。通过下面的测试，发现通过使用jy901自带的获得的传感器角度漂移较小，因此方案采用jy901的旋转角度进行三维手指模型控制。

（在整个角度稳定性验证的过程中，主要是涉及到几个方面。1、传感器不运动的状态下，查看立方体的运动状态。在人眼能识别的晃动情况下，发现立方体是静止状态，几乎没有晃动 2、晃动传感器从一个状态到另外一个状态，再回到原状态，查看立方体的运动情况。立方体能回到原来的位置状态）

**（3）上面方案选择，需要解决的问题**

**一方面是加速度数据误差导致二次积分距离出现很大偏差问题。这里面主要考虑传感器固定偏差和各种噪声对传感器数据的干扰问题。ZUPT的解决方案是检测脚掌的静止状态，对静止状态的数据进行处理。**

**一方面是因为陀螺仪传感器数据漂移问题导致积分出来的角度会有漂移。主要考虑陀螺仪传感器带来的数据漂移问题。**

1. **基于jy901的数据手套的设计**
2. **首先考虑多传感器数据读取采样频率**

传感器数据采集主要需要关注的几个点分别为：数据的精度问题、传感器的采样频率问题。对于JY901采集到的角度数据可以精确到小数点后5位，传感器采样频率为200HZ（本实验中直接读取寄存器中的roll yaw pitch角度数据，不涉及到积分问题，所以这里对传感器的采样频率要求可以降低，达到人眼能识别的频率以上就行），符合要求。

因为arduino本身处理器比较小或者其他原因，使用复用器进行多传感器数据读取，导致传感器采样频率降低。经测试，当传感器数目为11时，采样频率在30左右。人眼识别图像的速度为24帧，所以此时的采样频率已经满足要求。

1. **怎么使用arduino实现多传感器数据读取以及连线问题（复用器的使用）**

硬件需求：11个IMU传感器、2个I2C复用器、1个微处理器Arduino

**复用器引脚介绍：**图示



**复用器多路复用原理：**

通过三个数字脚分别输出相应的高低电位组合为8个不同的状态，可供8个传感器复用。

“时分复用”：同一时间只允许一路占用复用器。

**复用器本身没有在硬件层面集成对多路复用的控制功能，即没有专门的附加芯片通过时间片轮转的方式去分配时间的复用——多路复用的控制不在复用器端**

**多路复用的控制是用户在微控制器上通过程序实现的。**

**通过程序控制有两种实现方式：**

1. **时间片轮转，对选定的复用地址（比如选定0、1、2、3地址） 通过时间片轮转的方式分配占用时间。**
2. **按需调用，比如:当前时刻需要使用“1”地址，那么调用“地址选择函数”（实现对3个数字脚输出电位的控制），将复用器分配给与地址“1”相连的传感器使用，占用时间直至单次操作完成，下一时刻将复用器分配给另一个地址使用——当前的使用方式**

**进一步高效复用：**

1. 寻找 频分复器
2. 寻找 在硬件层面通过附加芯片集成复用控制功能的复用器

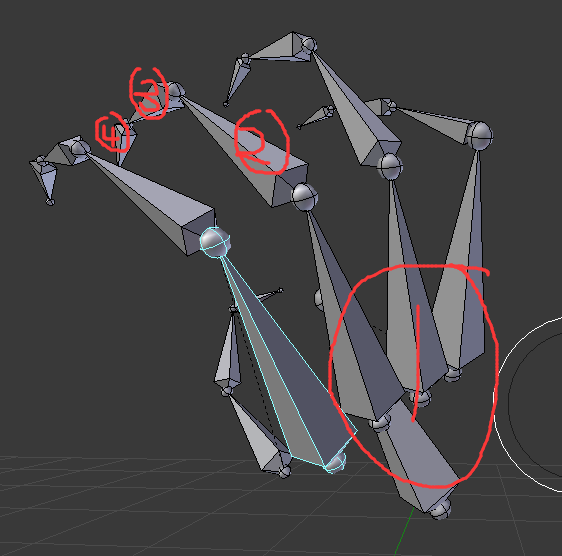
**硬件上复用器的使用：**

1. 复用器数量：每个复用器可提供8传感器连接，因此11个传感器需要2个复用器
2. 复用器使用方式：**级联，** 复用地址数量 ，n为级数。2个复用器可以构成不满的两级（9个复用器可构成满两级的结构，8个复用器分别接到1个复用器的8个地址，供64个传感器连接），可连接15个传感器（两个复用器连接占用一个地址）。
3. 多复用器之间连线：复用器间的连接，只和复用器地址的扩展有关，所以两个复用器的连接：一个复用器作为第一级，一个复用器作为第二级，选定第一级复用器的某一地址与第二级复用器的”Z”脚连接，将第一级复用器的一个地址扩展为第二级复用器的8个地址。
4. 复用器与传感器、微处理器连线：图示



数据手套硬件集成注意：

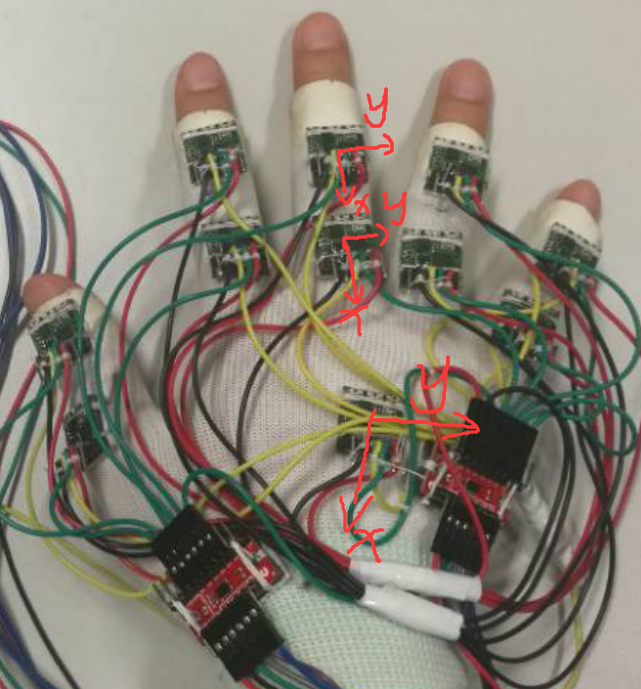
1. 确保传感器集成后手指运动的灵活性，在指关节处传感器引线的连接方式和长度、布局方面需要考虑；
2. 防止裸线之间接触连通，在连线的焊接和布局时方面需要考虑；
3. 传感器在数据手套上既要固定牢固，又要方便拆换；
4. 数据手套的制作不能损伤传感器元件，尽量不将双面胶直接粘在传感器上，影响传感器在其他场合下的再次使用；
5. 手套比较薄的情况下，防止传感器焊脚穿透手套与皮肤直接接触，两方面考虑：1防止手部出汗的情况下导通传感器的不同引脚，2防止手部干燥的情况下手部静电击坏传感器。可以通过皮革来隔离手套与传感器，同时皮革有一定的弹性缓冲作用，可以起到更好的固定作用。
6. **进行arduino程序书写**
7. **首先了解blender手掌模型中，如何控制手指骨架运动。**



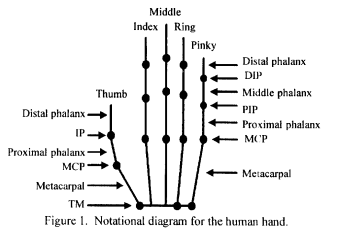
当设置1角度，其他关节旋转角度不做处理的情况下，整个手都是在绕着整个手掌的转动而转动，此状态的手掌状态是正确的；当设定2的角度为50时，控制1的旋转角度时，无论1的角度怎么变化，都会发现关节1和关节2的夹角永远是呈50度角。

从上面的测试可以推算出，每个关节只要获得和前一个关节的夹角数据，就能正确的对手指关节进行控制。

并且了解到IMU传感器输出的旋转角度是导航坐标系下的角度数据。通过手掌以及手指关节上面的各个IMU传感器能直接接收roll yaw pitch 角度。



左图所示，每个传感器的放置姿态都是一致的（X轴指向手心的方向，y轴指向外侧，z轴垂直于每个关节向上）。因此对于手指的弯曲状态可以使用y轴的角度的偏差进行控制，使用z轴角度的偏差控制MP关节的外展角度。



1. **使用数学公式对手掌模型进行阐述（先进行处理大拇指之外的模型进行阐述，下面的公式以食指数据为例 index表示食指，middle表示中指，ring表示无名指）**

首先水平张开手掌，采集手掌的静止数据。（这种做法是因为数据手套在制作的时候，z轴的朝向有可能具有偏差。静止状态下能记录初始传感器的偏差，在一定程度上纠正因为手套制作带来的手指跟踪误差。

采集10秒钟的静止数据，取传感器差值平均值作为手套误差数据：

大拇指角度控制：（其中，X表示从传感器读取的X轴的加速度数据，x表示传递给模型的角度数据）

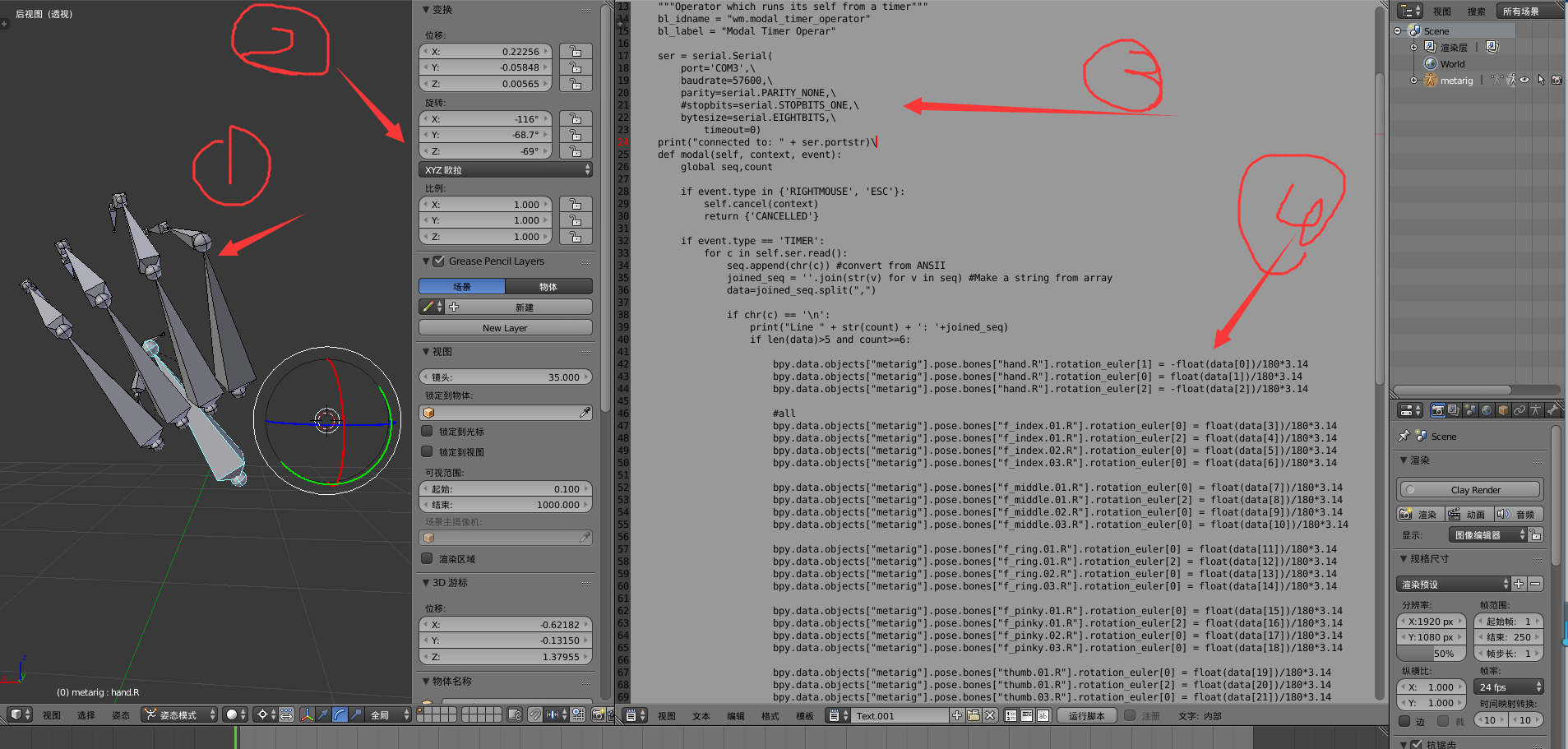
食指控制角度公示：

通过上面得到的角度值，在加上手指约束条件：

使用上面的公式进行手指角度约束。当计算出来手指角度大于上面约束的值时，则将约束值赋给它。

将得到的手指关节数据Serial.print()进行串口读取。

1. **Blender读取传感器数据，进而对手掌三维模型进行控制。（2018-03-21为文件夹下对应的文件）**



第一个区域是手掌模型。手掌模型使用的是blender库里面的三维模型；区域二是表示三维模型的属性区域，上面有旋转角度属性（属性上面可以选择使用的坐标系类型，以及每个关节的名字）；区域三指示串口的设置，上面设置串口的端口号、波特率等相关信息；区域四数据的处理阶段，从串口中读取数据，实时控制手指关节进行运动。