

# GitHub 下载题目与上传答案

[GitHub 仓库链接] <https://github.com/Songjiaxin11/Interview>

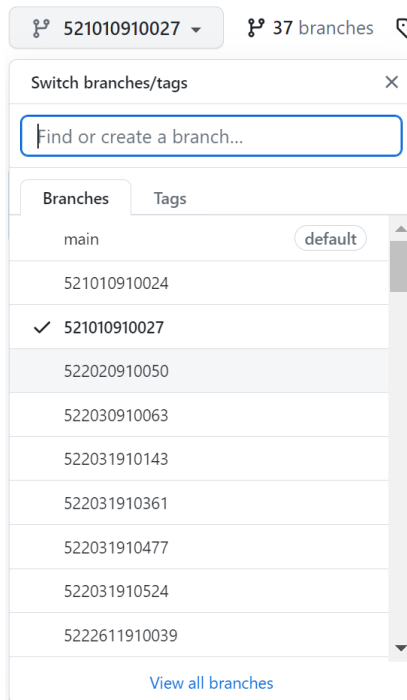
## 下载方式：

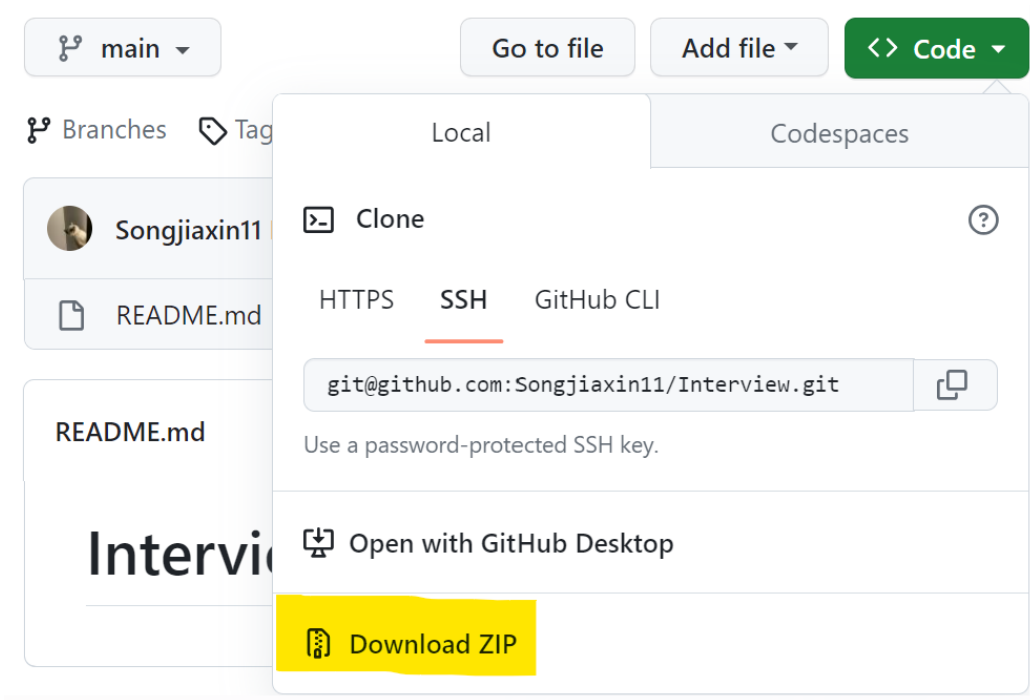
### 1. Git 指令

1. `git clone git@github.com:Songjiaxin11/Interview.git`
2. 找到自己的分支 `git checkout branchname`

### 2. Download Zip

1. 在 main 标志的小三角下拉 **先找到自己的分支, 点击切换**
2. 点击 Code 点击 Download Zip





## 上传方式:

### 1. Git指令 (15 points)

```
git add .  
git commit -m "message"  
git push
```

### 2. 压缩包上传至邮箱 (0 points)

owl.cq@sjtu.edu.cn

# C++基础 & VEXCode基础使用

## Eigen库介绍

Eigen是一个C++模板库, 提供了线性代数的基本功能, 包括矩阵, 向量, 矩阵分解, 矩阵函数等等. 以下是所有你可能用到的方法.(**不一定全部用到**)

- 定义一个向量 **Vector2d** 对象

$$v = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} 1.0 \\ 2.0 \end{bmatrix}$$

```
Vector2d v(1.0,2.0);  
//OR  
Vector2d v;  
v << 1.0, 2.0;
```

- 访问向量中的元素

```
v[0] = 1.0;  
v[1] = 2.0;  
//OR  
v.x() = 1.0;  
v.y() = 2.0;
```

- 取模

$$|v| = \sqrt{v_x^2 + v_y^2} = \sqrt{1.0^2 + 2.0^2} = \sqrt{5.0}$$

```
v.norm();
```

- 点乘

$$v_1 = \begin{bmatrix} 4.0 \\ 3.0 \end{bmatrix}$$

$$v_2 = \begin{bmatrix} 1.0 \\ 2.0 \end{bmatrix}$$

$$v_1 \cdot v_2 = 4.0 \times 1.0 + 3.0 \times 2.0 = 10.0$$

```
v1.dot(v2);
```

- 求角度

$$\theta = \arctan \frac{v_y}{v_x} = \arctan \frac{2.0}{1.0} = 1.107$$

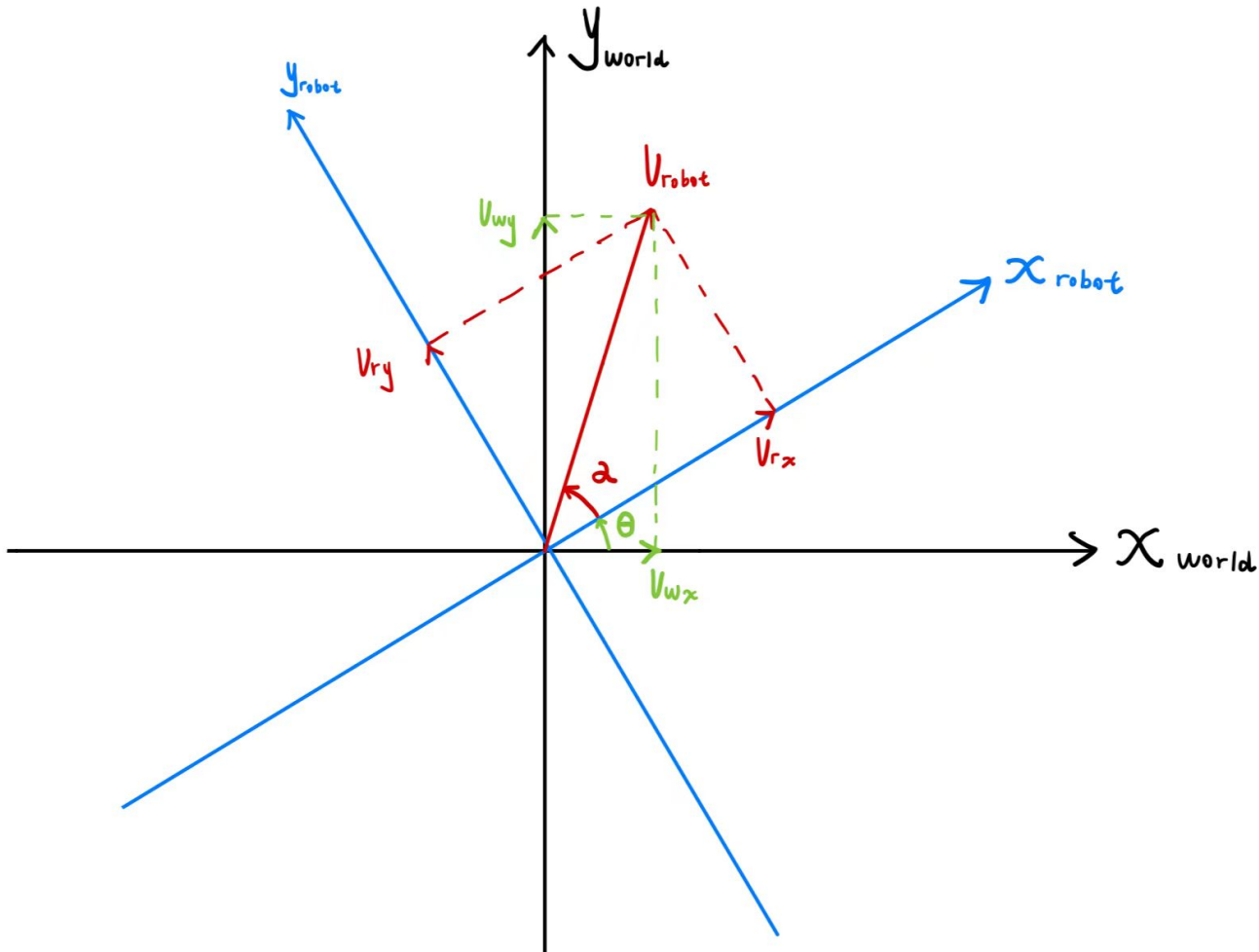
```
atan2(v.y(), v.x());
```

下图是坐标系转换示意图

下图是坐标系转换示意图

黑色代表世界坐标系, 蓝色代表机器人坐标系, 红色代表机器人坐标系下的速度, 绿色代表世界坐标系下的速度

接下来的任务中, 你需要实现从从机器人坐标系下速度到世界坐标系下速度的转换



## Task 1. (5 points)

由于部分学院C++课程采用英文授课, 此代码采用中英双语注释, 内容完全相同

在你面前是一个brain(主控)和一个inertial sensor(陀螺仪), 请在 robot-config.cpp 中调整相应的端口

```
inertial Inertial = inertial(PORT1);
```

## Task 2. (45 points)

chassis.h 中有一个 Chassis class , 其中说明了你需要实现的函数, 请在 chassis.cpp 中实现这些函数  
具体分值与要求已在 chassis.cpp 中明确

## Task 3. (50 points in total)

### 在 main.cpp 中的 int main() 函数中, 实现以下任务

以下是你可能用到的VEXCode函数(不一定全部用到)

```
//陀螺仪的初始化
void calibrate();
bool isCalibrating();
//陀螺仪的读数
double heading();
double rotation();
//设置打印光标位置
Brain.Screen.setCursor(1, 1);
//打印
Brain.Screen.print("Hello World");
```

### 具体要求在代码中已有详细说明, 你无需再继续阅读这份文档

#### 3.1: (30 points)

- (5 points) 初始化陀螺仪
- (5 points) 创建一个名为 myChassis 的 Chassis 类对象, 使用构造函数将初始化, 其中参数为 initial\_velocity
- (6 points) 得到机器人x方向和y方向的速度并在brain的屏幕上输出  
形式: Current Velocity: x: 10, y: 10
- (5 points) 使用 new 分配内存创建一个Vector2d\* 类型的指针, 名为 velocity\_Ptr , 将其初始化为  $v_x = 20, v_y = 20$
- (5 points) 使用 updateTrans() , 更新 myChassis
- (4 points) 输出( cout )更新后的速度
- 形式: Updated Velocity: x: x\_value, y: y\_value

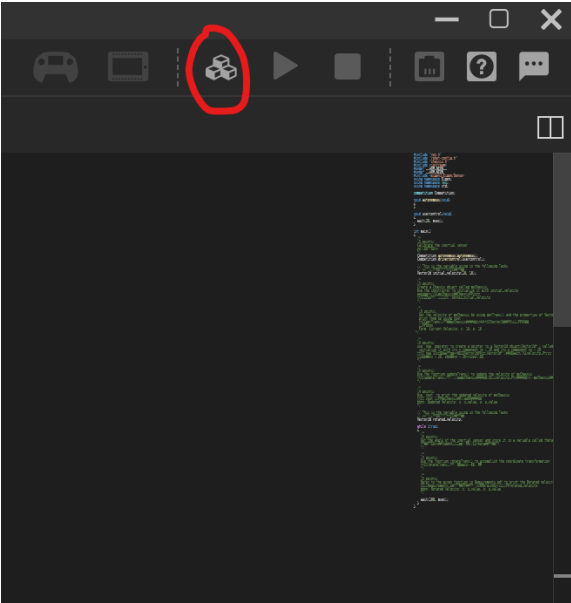
#### 3.2: (20 points)

- (5 points) 获取当前机器人旋转角度
- (10 points) 使用坐标系转换, 得到世界坐标系下机器人x,y方向的速度

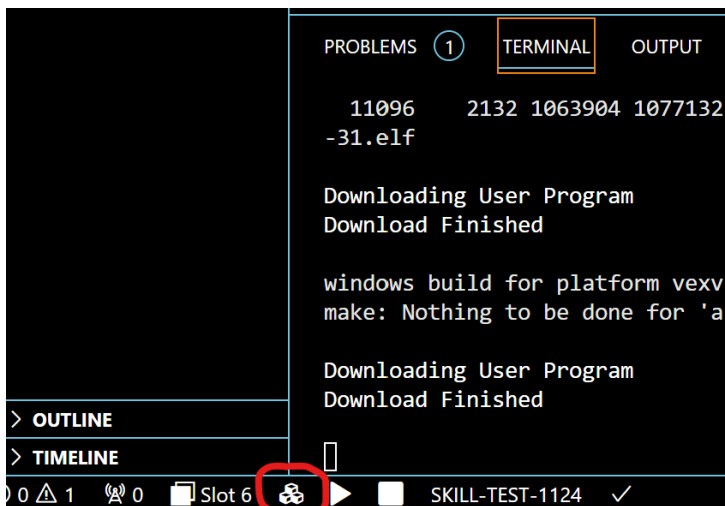
- (5 points) 在brain的屏幕上输出  
形式: Rotated Velocity: x: x\_value, y: y\_value

## 编译文件

1. 在VEXcode V5 Pro中点击右上角编译



2. 在VSCode中点击右下角编译



## 运行文件

使用Micro-USB连接电脑与brain, 当上图标志变为下载时, 点击下载即可导出程序