

```

#include "robot-config.h"
#include <iostream>

int apexSpeedPCT = 50;           //global vars
int armParallelSpeedPCT=100;
int armParallelPositionDEG=45;
int armExtendedSpeedPCT=100;
int armExtendedPositionDEG=135;
int manualArmSpeedPCT=80;
int armAutoReturnSpeedPCT=100;

int step=-1;                     //for auton motor
    position seeking
double seek=-1;
double currentPos=0;
double speed=0;

//competition object
vex::competition Competition;

void pre_auton( void ) {
}

void autonomous( void ) {

}

/*
Brings arm up, then brings it down. (arm goes higher up w/ more
negative numbers)
Enabled w/ usercontrol
*/
void armcontrol(void) {
    ApexMotor1.startRotateTo(-250,vex::rotationUnits::deg, 100,
        vex::velocityUnits::pct);//up
    vex::task::sleep(500);
    ApexMotor1.startRotateTo(-100,vex::rotationUnits::deg, 100,
        vex::velocityUnits::pct);//down
    vex::task::sleep(300);
    ApexMotor1.startRotateTo(0, vex::rotationUnits::deg, 25,
        vex::velocityUnits::pct);//down slower
    vex::task::sleep(300);
}

```

```

/*For opcontrol/ non-autonomous purposes*/
void usercontrol( void ) {
    //Enable arm
    armcontrol();

    //Drive Control: Arcade Mode
    while (1) {
        LeftMotor1.spin(vex::directionType::fwd,
            (Controller1.Axis3.value() + Controller1.Axis1.value()/2),
            vex::velocityUnits::pct); //(Axis3+Axis1)
        LeftMotor2.spin(vex::directionType::fwd,
            (Controller1.Axis3.value() + Controller1.Axis1.value()/2),
            vex::velocityUnits::pct); //(Axis3+Axis1)
        LeftMotor3.spin(vex::directionType::fwd,
            (Controller1.Axis3.value() + Controller1.Axis1.value()/2),
            vex::velocityUnits::pct); //(Axis3+Axis1)
        RightMotor1.spin(vex::directionType::fwd,
            (Controller1.Axis3.value() - Controller1.Axis1.value()/2),
            vex::velocityUnits::pct); //(Axis3-Axis1)
        RightMotor2.spin(vex::directionType::fwd,
            (Controller1.Axis3.value() - Controller1.Axis1.value()/2),
            vex::velocityUnits::pct); //(Axis3-Axis1)
        RightMotor3.spin(vex::directionType::fwd,
            (Controller1.Axis3.value() - Controller1.Axis1.value()/2),
            vex::velocityUnits::pct); //(Axis3-Axis1)

        vex::task::sleep(20); //Prevents wasted resources.
    }
}

int main() {
    pre_auton();

    //Set up callbacks for autonomous and driver control periods.
    Competition.autonomous( autonomous );
    Competition.drivercontrol( usercontrol );

    //Prevent main from exiting with an infinite loop.
    while(1) {
        vex::task::sleep(100);//Sleep the task for a short amount of
            time to prevent wasted resources.
    }
}

```