

File: C:\Users\grant\Documents\3946X\3946X-2018-19\3946x.c

```
#pragma config(Sensor, in1,    leftLift,        sensorPotentiometer)
#pragma config(Sensor, in2,    rightLift,       sensorPotentiometer)
#pragma config(Sensor, in3,    rotatorPot,     sensorPotentiometer)
#pragma config(Sensor, in4,    clawPot,        sensorPotentiometer)
#pragma config(Sensor, in5,    gyro,           sensorGyro)
#pragma config(Sensor, dgt19,  leftDriveQuad,  sensorQuadEncoder)
#pragma config(Sensor, dgt111, rightDriveQuad, sensorQuadEncoder)
#pragma config(Motor,  port2,    topLift,        tmotorVex393_MC29, openLoop)
#pragma config(Motor,  port3,    slingshot,      tmotorVex393_MC29, openLoop)
#pragma config(Motor,  port4,    frontRightDrive, tmotorVex393_MC29, openLoop)
#pragma config(Motor,  port5,    frontLeftDrive, tmotorVex393_MC29, openLoop)
#pragma config(Motor,  port6,    backRightDrive, tmotorVex393_MC29, openLoop)
#pragma config(Motor,  port7,    backLeftDrive,  tmotorVex393_MC29, openLoop)
#pragma config(Motor,  port8,    intake,         tmotorVex393_MC29, openLoop)
#pragma config(Motor,  port9,    claw,           tmotorVex393_MC29, openLoop)
#pragma config(Motor,  port10,   rotator,        tmotorVex393_HBridge, openLoop)

#pragma competitionControl(Competition)

//This file contains the basic structure of drive control and autonomous.

const int REDSIDE=1;
const int BLUESIDE=-1;

const int liftOutOfTheWayHeight=900;

const int fullPower=127;
const int driveThreshold=10;

const int liftLowPos=850;

const int clawOpenPos=1750;
const int clawClosePos=2300;

#include "Vex_Competition_Includes.c"
#include "functions.c"
#include "auton.c"

void pre_auton()
{
    calibrateGyro()

    drivePID.pGain=0.2;
    drivePID.iGain=0;
    drivePID.dGain=0;

    liftPID.pGain=0.125;
    liftPID.iGain=0;
    liftPID.dGain=0;

    gyroPID.pGain=0.25;
    gyroPID.iGain=0.0;
```

File: C:\Users\grant\Documents\3946X\3946X-2018-19\3946x.c

```
gyroPID.dGain=0;

clawPID.pGain=0.1;
clawPID.iGain=0;
clawPID.dGain=0;

rotatorPID.pGain=0.2;
rotatorPID.iGain=0;
rotatorPID.dGain=0;
}

task autonomous()
{
    nearAuton(BLUESIDE)
}

//Moves the lift to a height to not block the lifte
void getLiftOutOfTheWay() {
    if(liftPID.target<=liftOutOfTheWayHeight) liftPID.target=liftOutOfTheWayHeight;
}

task usercontrol()
{
    // Start subsystem tasks
    startTask(liftControl);
    startTask(rotatorTask);
    startTask(clawTask);
    startTask(driveLocker);
    rotatorPID.target=rotatorLowPos;
    clawPID.target=clawClosePos;
    while (true)
    {
        // Drive control
        if( abs(vexRT[Ch1]) > driveThreshold || abs(vexRT[Ch3]) > driveThreshold ){
            lockDrive=false;
            runLeftDrive(vexRT[Ch3]);
            runRightDrive(vexRT[Ch2]);
        }else{
            lockDrive=true;
        }

        // Shooter control
        if(vexRT[Btn8U]){
            motor[slingshot] = fullPower;
            getLiftOutOfTheWay()
        }else{
            motor[slingshot] = 0;
        }

        // Intake control
        if(vexRT[Btn7U]){
            motor[intake] = fullPower;
            getLiftOutOfTheWay();
        }else if(vexRT[Btn7D]){
            motor[intake] = -fullPower;
        }else{

```

File: C:\Users\grant\Documents\3946X\3946X-2018-19\3946x.c

```
    motor[intake] = 0;
}

// Rotator control
if(vexRT[Btn8LXmtr2]){
    liftPID.target+=500;
    wait1Msec(500);
    if(rotatorPID.target==rotatorLowPos) rotatorPID.target=rotatorHighPos;
    else if(rotatorPID.target==rotatorHighPos) rotatorPID.target=rotatorLowPos;
}

// Claw control
if(vexRT[Btn7RXmtr2]){clawPID.target=clawOpenPos;clawIdle=false;}
else if(vexRT[Btn7LXmtr2]){
    //open the claw less when the rotator is in the low position
    if(rotatorPID.target<200) clawPID.target=clawClosePos-200;
    else clawPID.target=clawClosePos;

    clawIdle=false;
}
if(vexRT[Btn8UXmtr2]) rotatorPID.target=rotatorHighPos;
else if(vexRT[Btn8DXmtr2]) rotatorPID.target=rotatorLowPos;

if(vexRT[Btn7DXmtr2]) clawIdle=true;
}
}
```

