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, dgtl9,  leftDriveQuad,  sensorQuadEncoder)
#pragma config(Sensor, dgtl11, rightDriveQuad, sensorQuadEncoder)
#pragma config(Motor,  port2,           topLift,       tmotorVex393_MC29, openLo
#pragma config(Motor,  port3,           slingshot,     tmotorVex393_MC29, openLo
#pragma config(Motor,  port4,           frontRightDrive, tmotorVex393_MC29, oper
#pragma config(Motor,  port5,           frontLeftDrive, tmotorVex393_MC29, openl
#pragma config(Motor,  port6,           backRightDrive, tmotorVex393_MC29, openl
#pragma config(Motor,  port7,           backLeftDrive, tmotorVex393_MC29, openLo
#pragma config(Motor,  port8,           intake,        tmotorVex393_MC29, openLo
#pragma config(Motor,  port9,           claw,          tmotorVex393_MC29, openLo
#pragma config(Motor,  port10,          rotator,       tmotorVex393_HBridge, ope
//*!!Code automatically generated by 'ROBOTC' configuration wizard

// Select Download method as "competition"
#pragma competitionControl(Competition)

//Main competition background code...do not modify!
#include "Vex_Competition_Includes.c"
#include "functions.c"
#include "auton.c"



void pre auton()
{
  calibrateGyro()
}



task autonomous()
{
  //-1 for blue
  //1 for red
  farAuton(-1);
}


void getLiftOutOfTheWay(){
  if(liftPID.target<=850)liftPID.target=850;
}
int liftLowPos=850;

int clawOpenPos=2850;
int clawClosePos=2200;

task usercontrol()
{
  startTask(liftControl);

// Start subsystem tasks
  startTask(rotatorTask);
  startTask(clawTask);
```

```
  rotatorPID.target=2900;
  clawPID.target=clawClosePos;
  while (true)
  {
    if(vexRT[Btn5d]){
      pTurn(-900);

    }
// Drive control
    if( abs(vexRT[Ch1]) > 20 || abs(vexRT[Ch3]) > 20 ){
      runLeftDrive(vexRT[Ch3]);
      runRightDrive(vexRT[Ch2]);
    }else{
      drive(0);
    }

// Shooter control
    if(vexRT[Btn8U]){
      motor[slingshot] = 127;
      getLiftOutOfTheWay()
    }else{
      motor[slingshot] = 0;
    }

// Intake control
    if(vexRT[Btn7U]){
      motor[intake] = 127;
      getLiftOutOfTheWay();
    }else if(vexRT[Btn7D]){
      motor[intake] = -127;
    }else{
      motor[intake] = 0;
    }

// Rotator control
    if(vexRT[Btn8LXmtr2]){
        liftPID.target=2300;
        wait1Msec(500);
        if(rotatorPID.target==rotatorLowPos)rotatorPID.target=rotatorHighPos;
        else if(rotatorPID.target==rotatorHighPos)rotatorPID.target=rotatorLowPo
    }

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


      clawIdle=false;

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


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

```
  }

  }
```