```
#pragma config(Motor,  port2,           leftMotor,     tmotorVex269_MC29, openLoop, reversed,
driveLeft)
#pragma config(Motor,  port4,           rightMotor,    tmotorVex269_MC29, openLoop, driveRight)
#pragma config(Motor,  port6,           liftMotor,     tmotorVex269_MC29, openLoop)
#pragma config(Motor,  port7,           servoPort,     tmotorServoStandard, openLoop)
#pragma config(Motor,  port9,           fireMotor,     tmotorVex269_MC29, openLoop)
//*!!Code automatically generated by 'ROBOTC' configuration wizard               !!*//

// LETZ GET THE HAX GOIN

/*
BookMarks:


IMPORTANT VARIABLES
Motor ports:
Left Motor: motor port 2
Right motor: motor port 4
Lift motor: motor port 9
Fireing pin motor: port 6

Controller Channels:
Lift channel: Channel 1
fire Channel: Channel 2
Servo Channel: Channel 8U and 8D

motor command syntax:
motor[port] = power; (power between 0-127)
setServo(serverPort, position); (position between -127 and 127)

CHANGELOG
Version 1.1
Fleshed out functions and control struccture.
Finished what should be the final program

Version 1.0
Began fundamental program setup, Outlined what is necessary for the robot to perform properly

*/



//Controller Ports
```

```
int lMotorCH = 2;
int rMotorCH = 1;

int servoPos = 0; //this will be adjusted in the code as needed
int lowSpeed = 4; // low speed will be 1/4 of high speed (or whatever number is used)
int speedy = 1; //if 0 robo will be set to low speed
int collectorSpeed = 0.2; // speed for the hazardous waste collector

int liftPos = 25; //lifted servo position
int lowPos = 125; // lowered servo position



task main()
{

	//basic syntax:
	/*
	define fireing sequence function
	define motor controls as functions for easier repetition
	define two different modes for high/speed to switch between speed and accuracy
controls

	if robot is in high speed mode:

	if left control is moved:
	move left motor at high speed (should be default vexRT[ch] joystick mapping values)

	if right control is moved:
	move right motor at high speed

	if robot is in low speed mode:

	if left control is moved:
	move left motor at low speed (vexRT[ch] mappping values divided by a limiter to reduce
max speed)

	if right control is moved:
	move right motor at low speed

	if fireing button is pushed:
	activate fireing sequence
```

```
	if lift control is pushed:
	move lift motor appropraite direction

	if speed control button is pushed:
	toggle between high and low speeds.
	*/

	while (true) { // Operating loop loop


		if (speedy == 1) { // if the motors are set to run faster for higher movement, run
the motors at max speed
			motor[leftMotor] = vexRT[lMotorCH];
			motor[rightMotor] = vexRT[rMotorCH];
			} else { // if the motors are set to run at a slower speed for precision, run
them at a predetermined reduced speed (full speed divided by the reducer variable, lowSpeed)
			motor[leftMotor] = vexRT[lMotorCH]/lowSpeed;
			motor[rightMotor] = vexRT[rMotorCH]/lowSpeed;
		}

		//Update servoPos based on if the buttons are pressed
		if(vexRT[Btn6U] == 1) {
			if (servoPos <= 127) { //don't over extend the servos
				servoPos = liftPos;
			}
		} else if(vexRT[Btn6D] == 1) {
			if (servoPos >= -127) { //don't over extend the servos
				servoPos = lowPos;
			}
		}

		motor(servoPort) = servoPos; //Repeatedly sets the servo to the servoPos
position

		// if fire button is pressed, activate fire function
		if (vexRT[Btn8D] == 1) {
			motor[fireMotor] = 127; //(motor may need to be reversed to fire the right
direction)
			} else if (vexRT[Btn8D] == 0) {
			motor[fireMotor] = 0;
		}
```

```
		// if the lift motor(for aiming the beam) is activated, execute appropriate motion
		if (vexRT[Btn5U] == 1) {
			motor[liftMotor] = 127; //raises the beam
			}	else if (vexRT[Btn5D] == 1) {
			motor[liftMotor] = -127; //lowers the beam
			} else {
			motor[liftMotor] = 0;		// lift speed to zero when not in use
		}

		// if switch button is pressed, toggle speedy between true and false
		if(vexRT[Btn7L] == 1) {
			speedy = 0;
		}
		if(vexRT[Btn7R] == 1) {
			speedy = 1;
		}


		// end while
	}


	// end main
}
```