Calibration of the distance traveled and steering sensors

	Rotation sensor used to measure distance traveled in inches		Touch sensor used to measure angles in degrees	
	distance	counts	angle	counts
Try 1	25.25	-121	139	97
Try 2	25.75	-122	146	96
Try 3	25.5	-121	146	97
Try 4	25.5	-122	149	97
Try 5	25.5	-121	151	96
Try 6	26	-124	152	96
Try 7	25.75	-122	153	96
Total	179.25	-853	1036	675
Counts per Inch		-4.758717	Counts per degree	0.651544

To use the calibration measure the distance the robot will travel and multiple by the counts per inch. This gives the number of counts that must be recorded to travel that distance. The count value is then used in a loop such as Wait Until. When the Wait Until loop reaches the stated counts the motor can be directed to stop or the robot can be directed to do some other function.

Similarly for the angle calibration. The angle the robot is wished to turn through is multiplied by the counts per degree. That count value is also used in Wait Until loops, or similar, until the desired angle is reached.

Procedure: For each sensor a small program was written that either travels a distance or turns for a specific amount of time, 2 or 5 seconds. For each trial the distance traveled or angle turned is measured and the count valuen is displayed on the RCX viewport. Several trials were done and the average counts per inch or degree was calculated.

```
program test_distance {

        #include <RCX2.h>
        #include <RCX2MLT.h>
        #include <RCX2Sounds.h>
        #include <RCX2Def.h>
        sensor rotation3 on 3
        rotation3 is rotation as angle

        main {
                ext InterfaceType "kFreestyle"
                rcx_ClearTimers
                bbs_GlobalReset([A B C])
                try {
                        rcx_Calibrate(4,4)
                        clear Rotation3
                        power [ C  ] 8
                        direction [ C  ] [ ]
                        on [ C  ]  for 200
                        display rotation3
                        stop tasks
                } retry on fail
        }


}
```

```
program test_rotation {

        #include <RCX2.h>
        #include <RCX2MLT.h>
        #include <RCX2Sounds.h>
        #include <RCX2Def.h>
        sensor touch2 on 2
        touch2 is switch as boolean
        event tPress_touch2EventPress when touch2.pressed



        main {
                ext InterfaceType "kFreestyle"
                rcx_ClearTimers
                bbs_GlobalReset([A B C])
                start TouchWatcher0
                rcx_Priority( 8)
                trigger tPress_touch2EventPress
                try {
                        counter1 = 0
                        direction [ B  ] [ ]
                        on [ B  ]  for 500
                        display counter1:1
                        stop tasks
                } retry on fail
        }


        watcher TouchWatcher0 monitor tPress_touch2EventPress
        {
                rcx_Priority( 3 )
                try {
                        counter1 += 10
                } restart on fail
        } restart on event
}
```