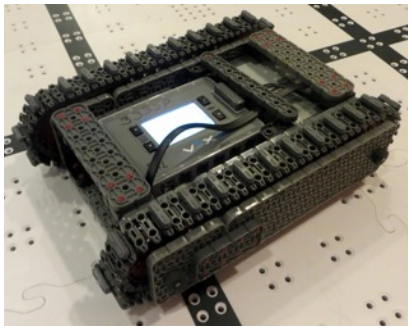
Tank Tread Drivetrain



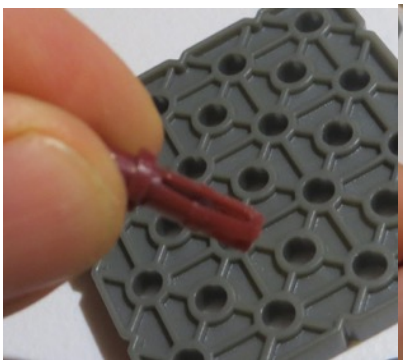
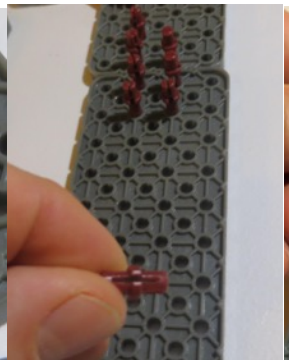
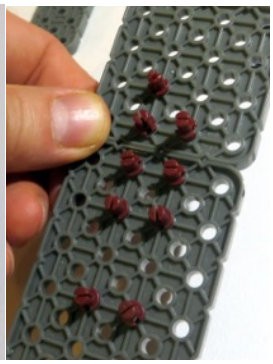
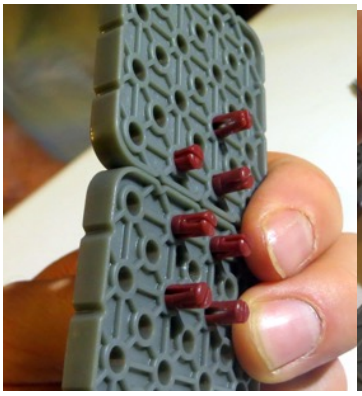
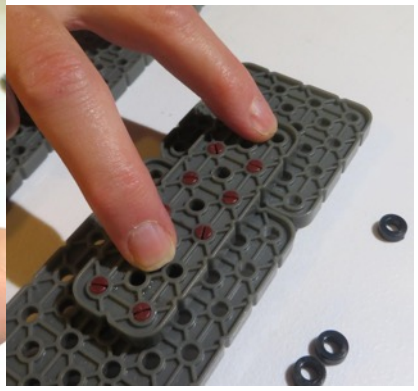
Before you start, make sure you have the right parts available. Set these parts on a table, and put all other parts away for now.

Brain & Battery	4 1x standoffs	4 2x2 black connectors	8 thin washers
Controller	12 4x standoffs	3 2x8 beams	about 4 ft of tank tread
2 motors	4 4x12 plates	2 1x8 beam	about 52 small nubs
4 24 tooth sprockets	4 #2 shafts	6 rubber washers	
2 2x6 beams	6 2x4 beams	4 4x4 plates	
2 smart cables	4 2x8 beams	24 spacer washers	
29 triple connectors	8 2x5 black connectors	handfull of single connectors	

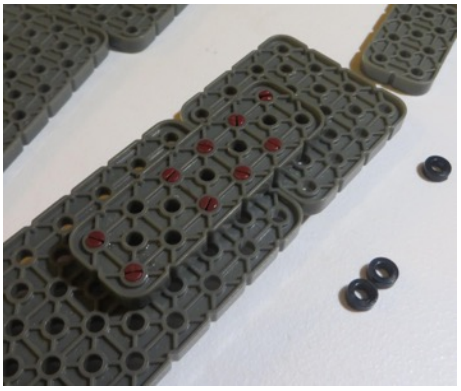
The Tank Tread drivetrain will consist of two sides that will be built separately and then attached together. If there are enough people helping with the build, then you could assign each side to a couple kids. The two sides are not identical, but mirror each other.

Left Drivetrain

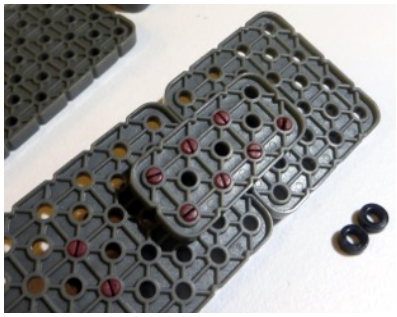
Step 1. Build first side support. Supplies: (1) 4x4 plate, (1) 2x12 plate, (7) triple connectors, (2) normal connectors, (1) 2x6 beam, (1) 2x4 beam

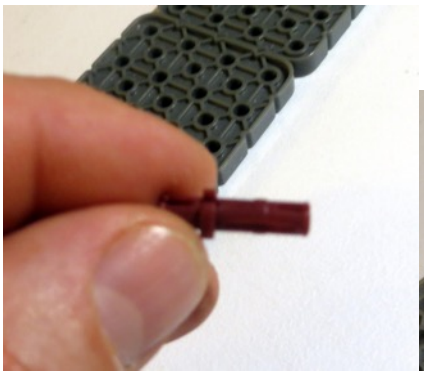
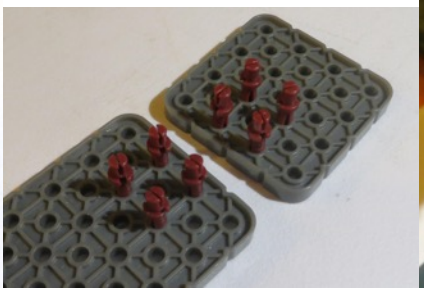
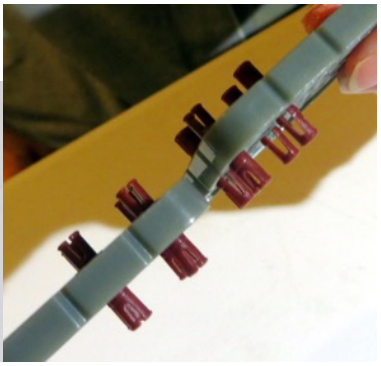
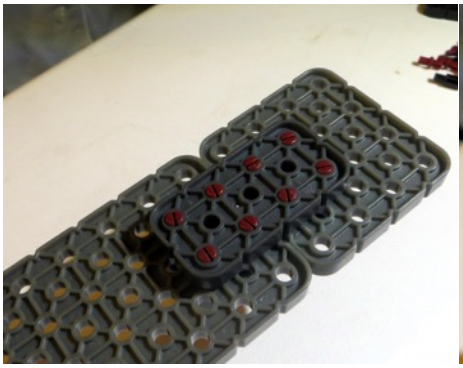
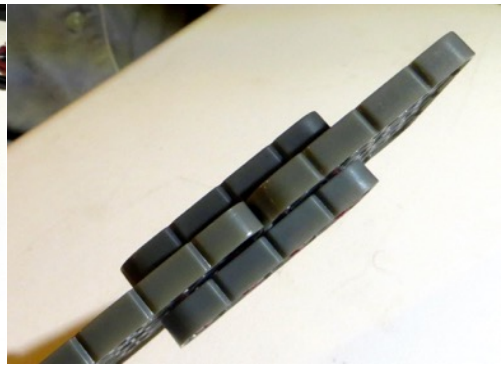
One side of the plate will look like this:



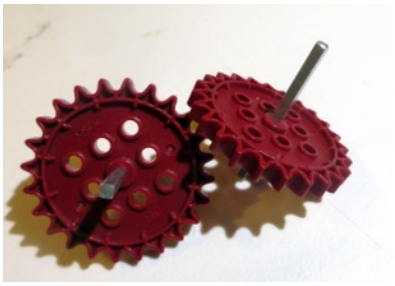
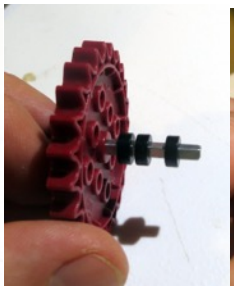
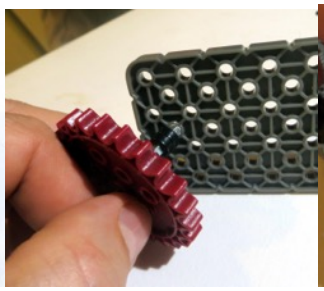
And the other side will look like this:

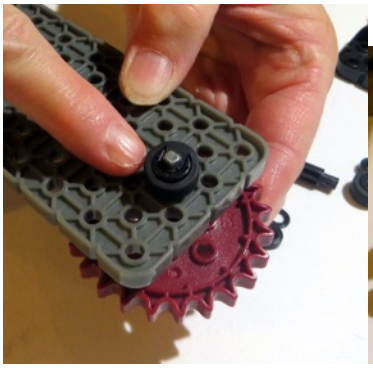
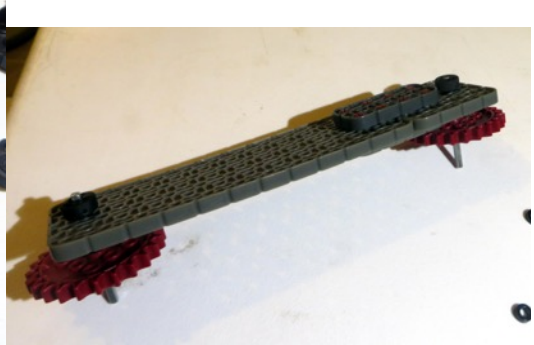


Step 2. Build second side support. Supplies: (1) 4x4 plate, (1) 2x12 plate, (8) triple connectors, (2) 2x4 beams

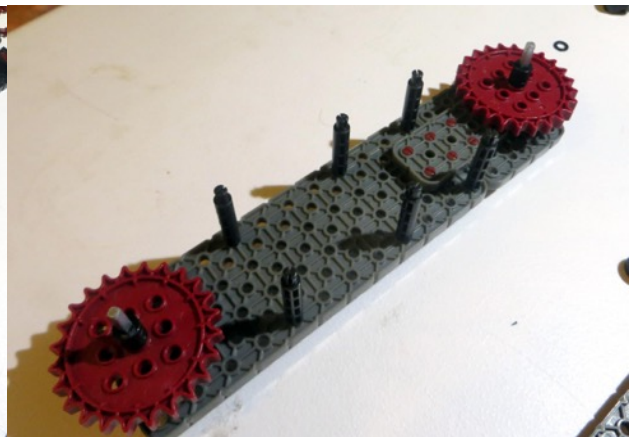
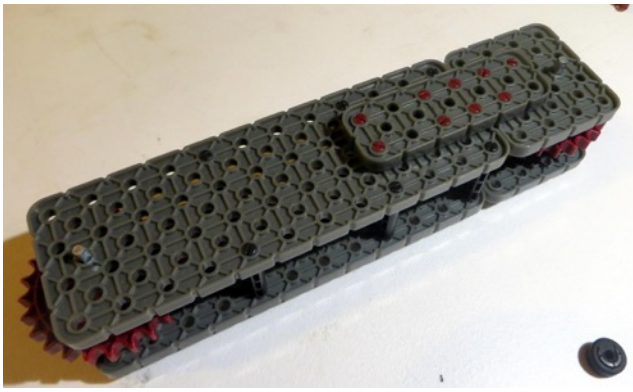
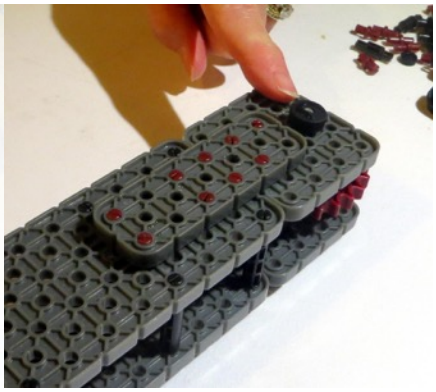
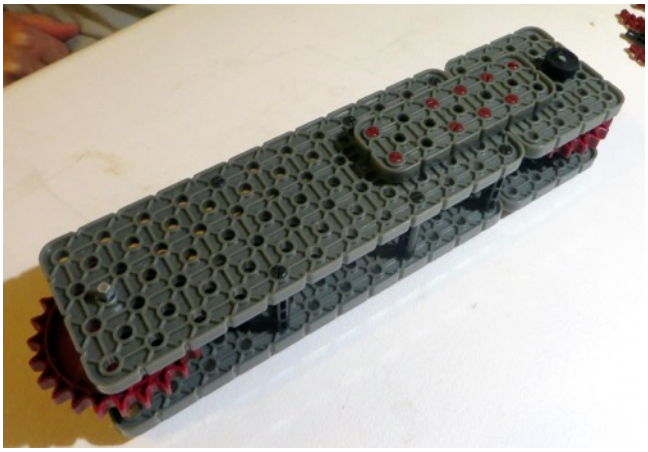






Step 3. Attach sprockets. Supplies: (2) medium sprockets, (6) spacer washers, (2) #2 shafts, (2) thin washers, (2) rubber washers

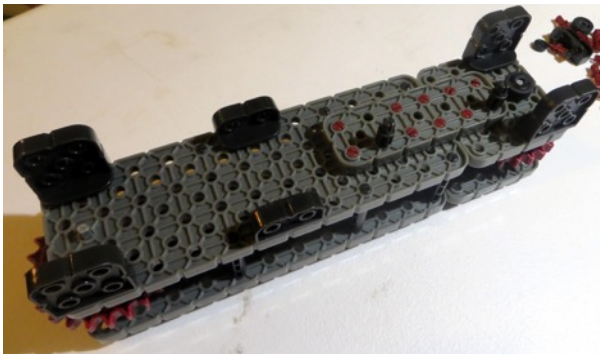




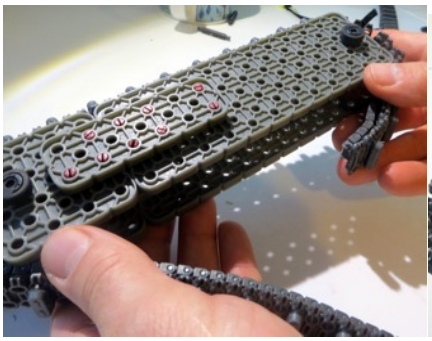
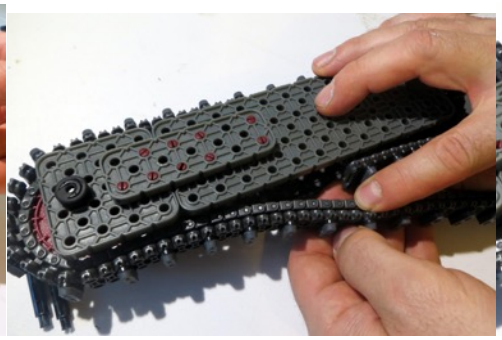
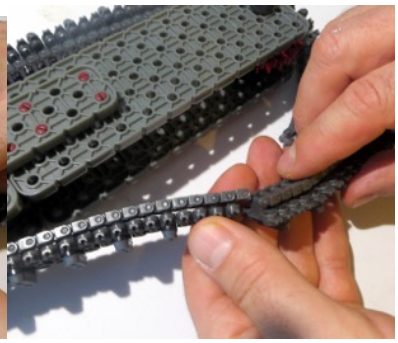
Step 4. Sprocket sandwich. Supplies: (6) 4x standoffs, (6) spacer washers, (1) thin washer, (1) rubber washer.

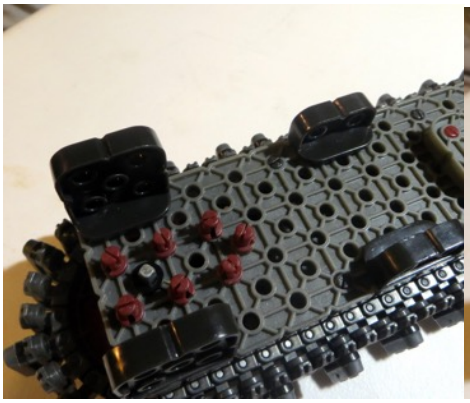
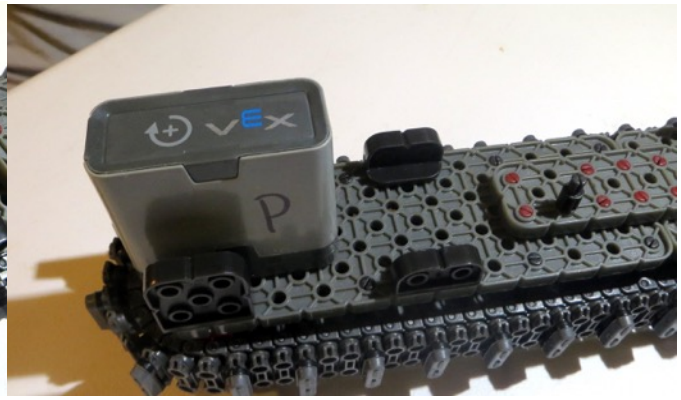
Step 5. Attach support brackets. (4) 2x5 support brackets, (2) 2x2 support brackets, (2) 1x standoffs



Step 6. Attach tank tread. Supplies: About 4 ft of tank tread with small numbs attached every third thread link

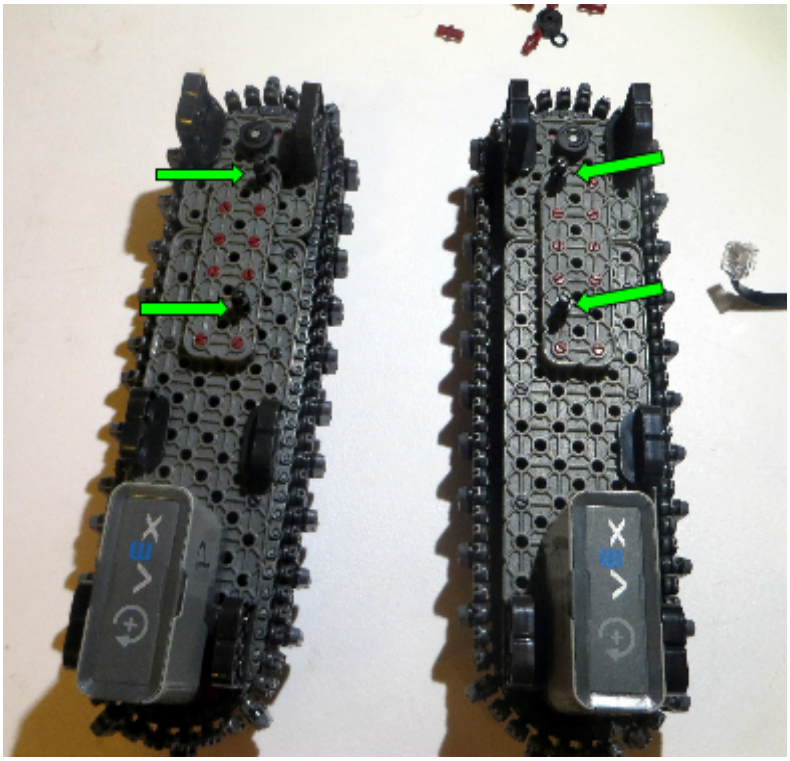




Step 7. Mount Motor. Supplies (7) normal connectors, (1) thin washer, (1) motor,

The Right Drivetrain

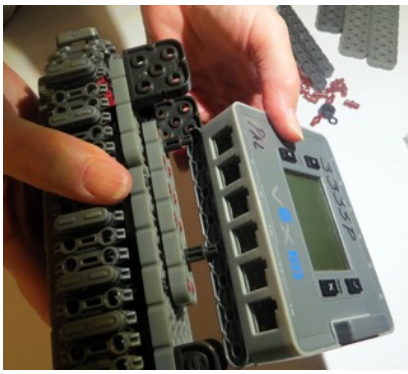
Same as the Left Drivetrain except for the location of the 1x standoffs that the green arrows point to

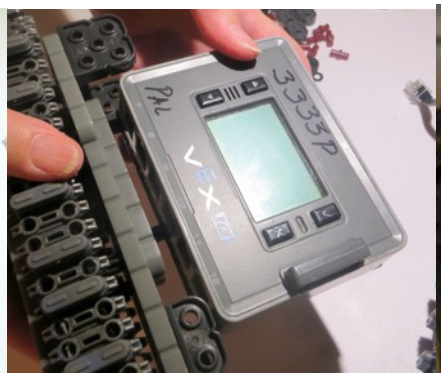


Hooking Up The Brain

Step 1. Plug smart cables into ports 6 and 7. Supplies: (2) short smart cables, (1) brain

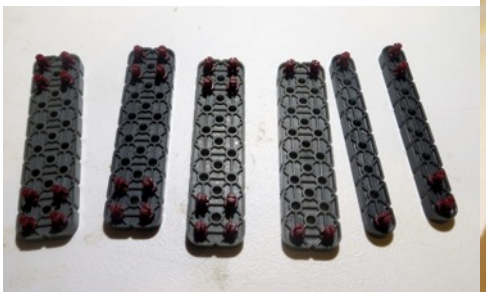
Step 2. Attach brain to left and right sides.

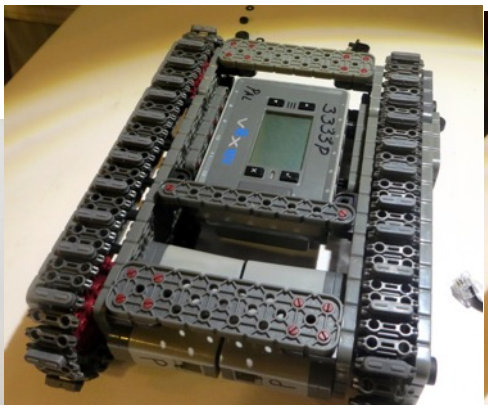


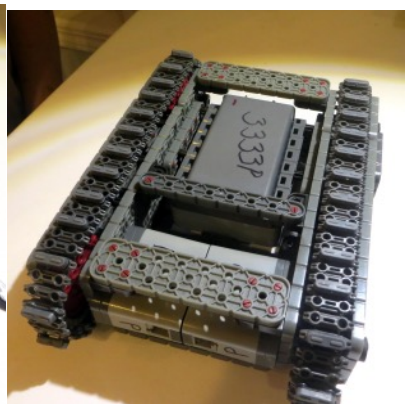


Attach Cross Supports

Supplies: (4) 2x8 beams, (2) 1x8 beams, lots of small connectors

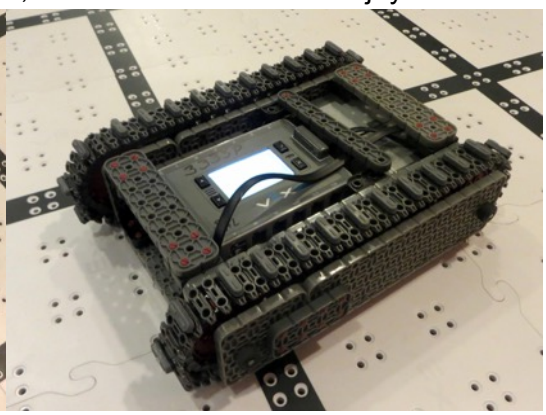






Running the Robot

Step 1. Turn on brain, select driver control,then select Run. Use the joystick controls.



#CODE Note that you can run the tank without this code, but the code will make it easier to drive
#Cut & Paste everyting below this line into robot c

```
#pragma config(Motor,  motor4,    leftFrontDriveMotor,        tmotorVexIQ, PIDControl,
        encoder)
#pragma config(Motor,  motor5,    rightFrontDriveMotor,       tmotorVexIQ, PIDControl,reversed, encoder)
#pragma config(Motor,  motor10,    leftRearDriveMotor,        tmotorVexIQ, PIDControl,
        encoder)
#pragma config(Motor,  motor11,    rightRearDriveMotor,       tmotorVexIQ, PIDControl,reversed, encoder)

//*!!Code automatically generated by 'ROBOTC' configuration wizard               !!*//

signed char limit_motor(signed int input)
{
        if(input > 100)
        {
                return 100;
        }
        else if (input < -100)
        {
                return -100;
        }
        return input;
}

//Cheesydrive tunable constants
#define SPEEDTURN_SENS 1.4   //these two constants should not be the same?
#define QUICKTURN_SENS 1.4
#define HALO_TURN_CONST 27   // cut off point for doing quick turn

//Function prototypes from Cheesydrive
void DoHaloDrive(signed char throttle, signed char wheel);

//Cheesydrive code (originally from cheesydrive.c) - I only copied in Halo since that's what we use
signed char pwmLeft,pwmRight = 0;
void DoHaloDrive(signed char throttle, signed char wheel)
{
        unsigned char quickTurn = 0;
        if(abs(wheel) > HALO_TURN_CONST)//If we should do SpeedTurn
        {
                quickTurn = 1;
        }//End If

        //Do the magic Cheesy Drive
        signed int diff;
        if(quickTurn)
        {
                //Add throttle to diff.
```

```
		//Wheel is diff
		diff = wheel * QUICKTURN_SENS;
		//Do diff
		pwmLeft = limit_motor((throttle + diff));
		pwmRight = limit_motor((throttle - diff));
	}//End QuickTurn
	else //If SpeedTurn
	{
		//We have to convert throttle to a float
		//Cap throttle at +-100
		throttle = limit_motor(throttle);
		float fthrottle = (float)throttle;
		//Pull it to a decimal number, -1 to 1
		fthrottle /= 100;
		//Diff = wheel * fthrottle * sens
		diff = (int)wheel * (float)fthrottle * SPEEDTURN_SENS;
		//Add and subtract Diff from each to get pwm
		pwmLeft = limit_motor((throttle + diff));
		pwmRight = limit_motor((throttle - diff));
	}//End SpeedTurn
}//End DoHaloDrive

task main()
{
	while(1)
	{
		//Do Halo Drive with the primary Y and primary X joysticks
		DoHaloDrive(getJoystickValue(ChA),getJoystickValue(ChC));

		setMotorSpeed(leftFrontDriveMotor,pwmLeft);
		setMotorSpeed(rightFrontDriveMotor,pwmRight);
		setMotorSpeed(leftRearDriveMotor,pwmLeft);
		setMotorSpeed(rightRearDriveMotor,pwmRight);

		//Wait the timing interval
		wait1Msec(25);
	}
}
```