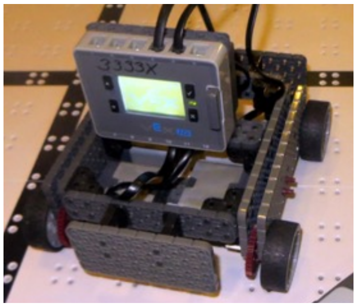
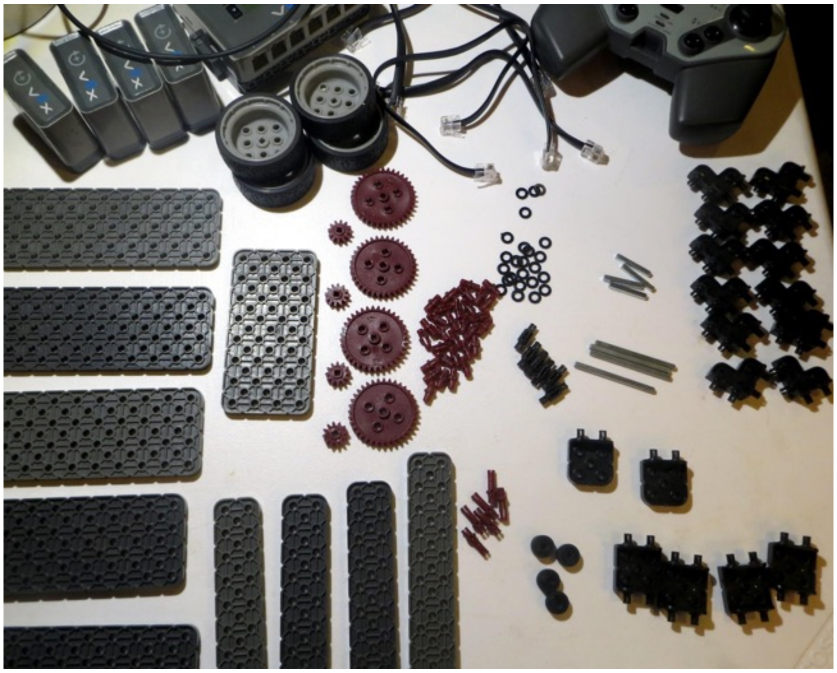
The Power Turtle Drivetrain





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.

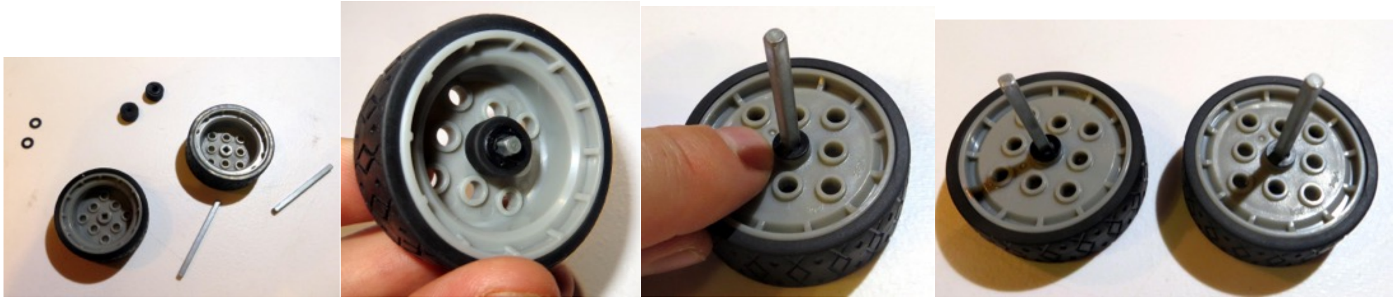
Brain & Battery	4 small gears	1 4x8 plate	4 #2 shafts
Controller	4 medium gears	3 2x8 beams	4 #1 shafts
4 motors	4 rubber washers	1 2x12 beam	12 4x3 corner black connectors
4 small wheel hubs	20 thin washers	5 4x12 plates	2 5x2 black connectors
4 small thin tires	handfull of single connectors	6 1x standoffs	4 4x5 black connectors
4 smart cables			

The Power Turtle 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 identical to each other.

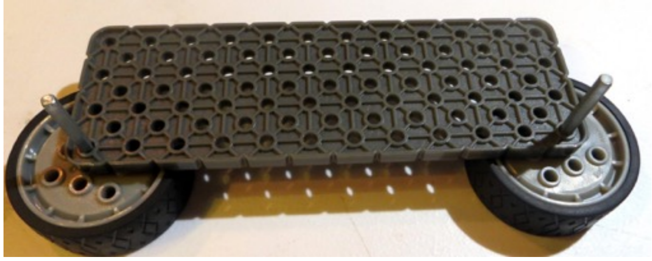
One Side of Drivetrain

Build two complete sides using this same set of instructions

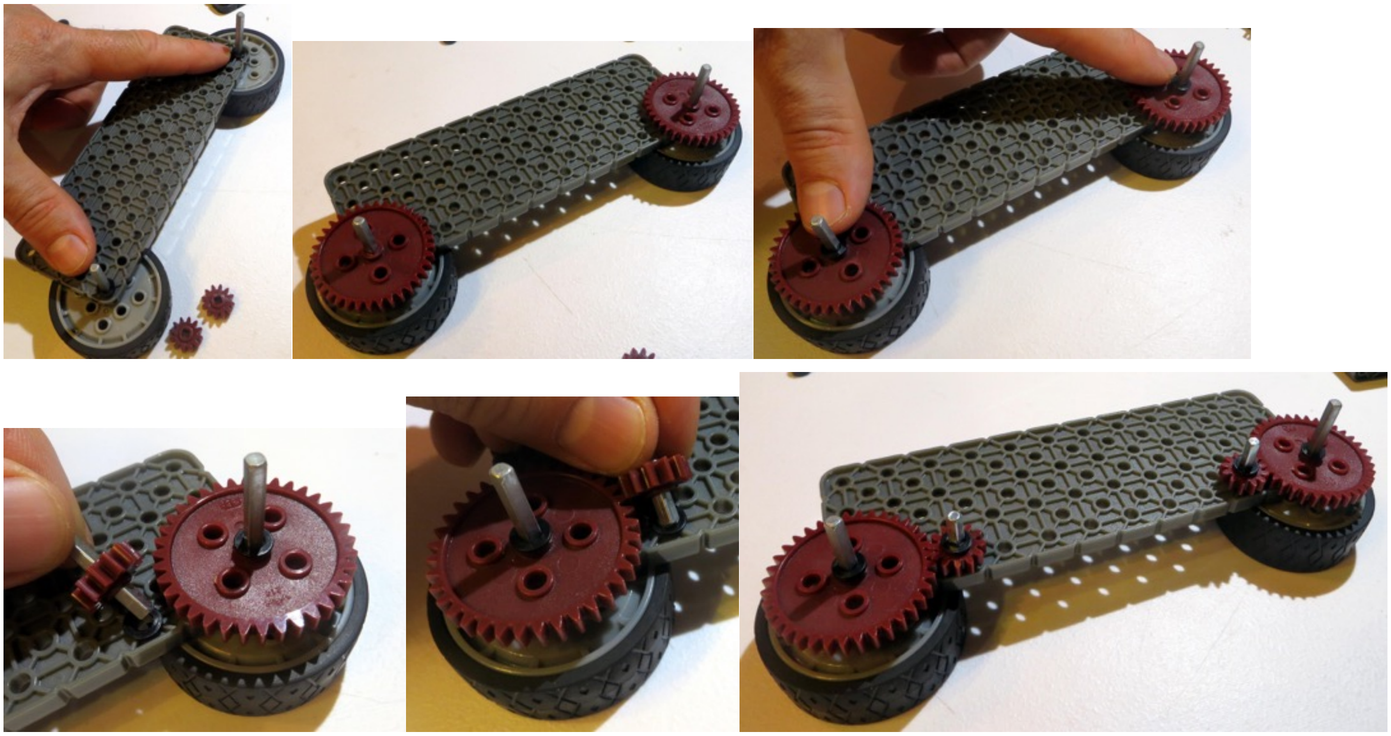
Step 1. Build wheels.   Supplies: (2) small hubs, (2) thin tires, (2) rubber washers, (2) #2 shafts, (2) thin washers



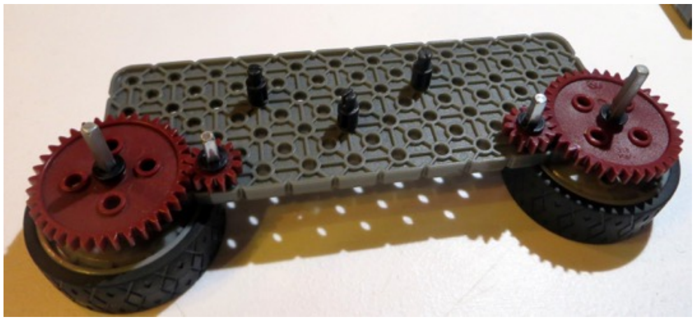
Step 2. Attach wheels. Supplies: (1) 4x12 plate

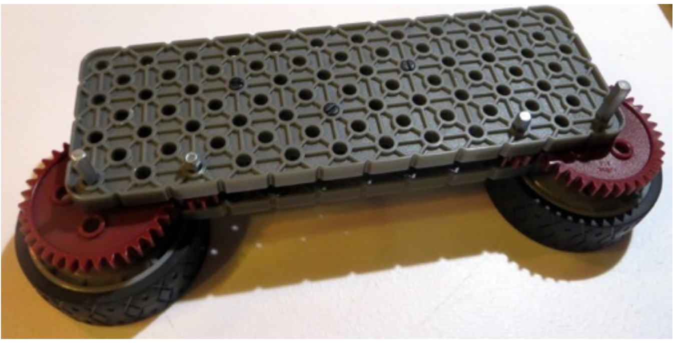


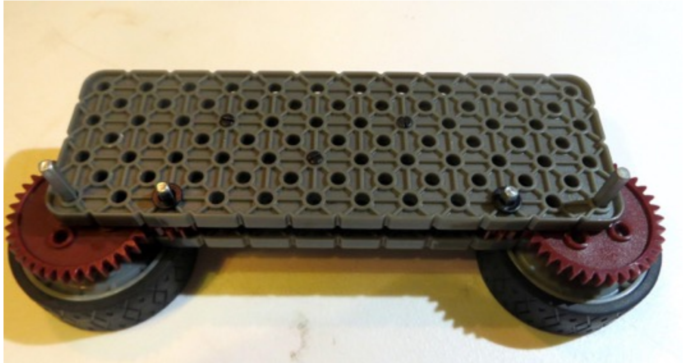
Step 3 Gear assembly. Supplies: (2) small gears, (2) medium gears, (8) spacer washers, (2) #2 shafts,



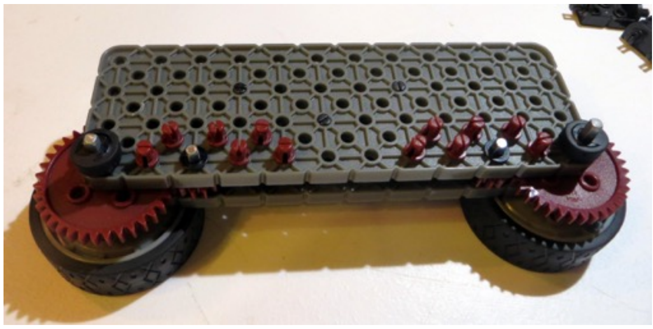
Step 4. Gear sandwich. (1) 4x12 beam, (3) 1x standoffs, (2) thin washers, (2) rubber washers

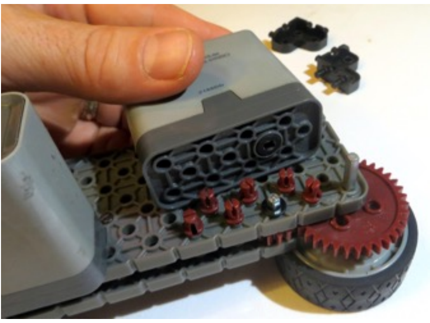


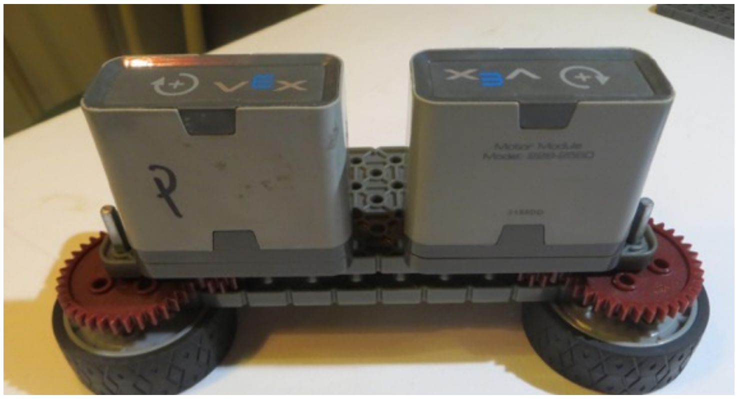




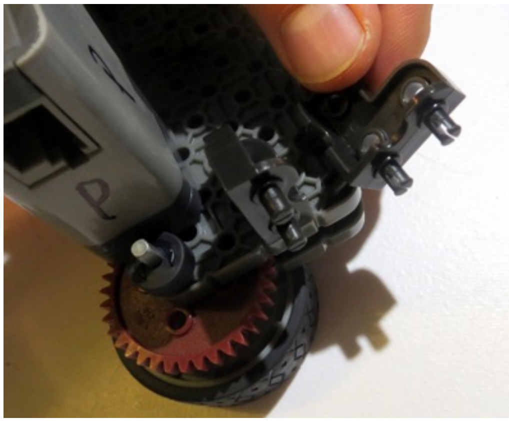
Step 5. Motor mount. (12) normal connectors, (2) thin washers, (2) motors

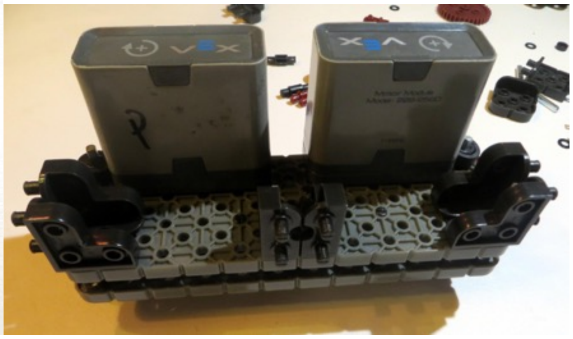


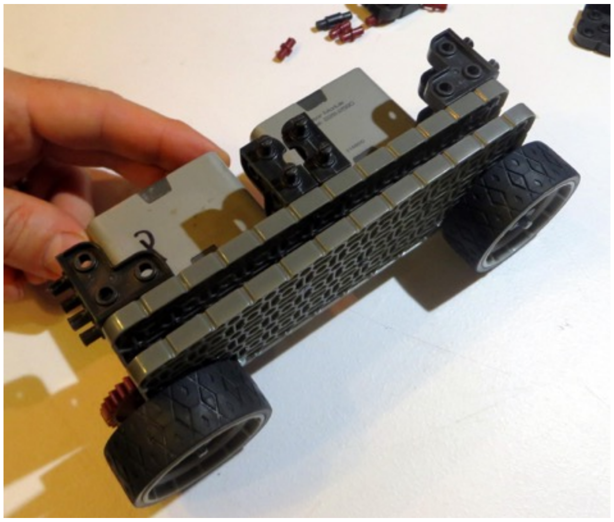




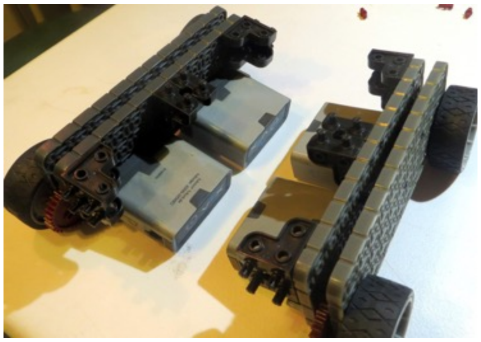
Step 6. Left support connectors. Supplies: (6) 4x3 corner brackets





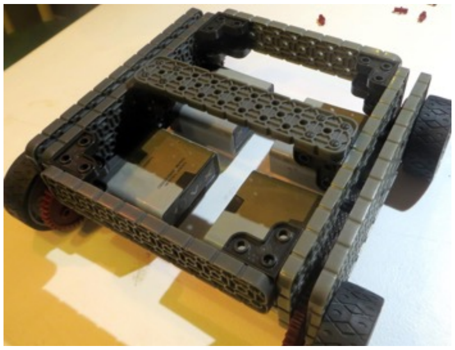


After the first side is built, build another one that is just the same.

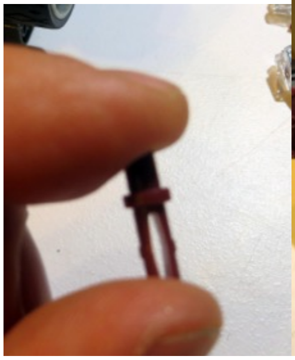
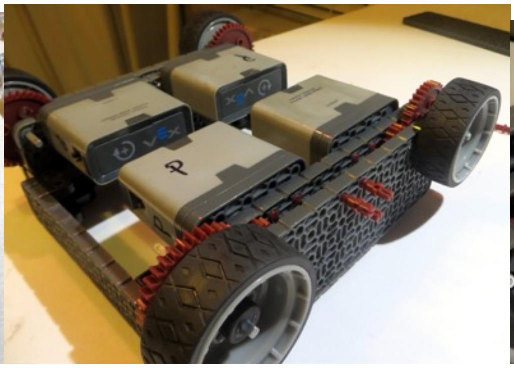
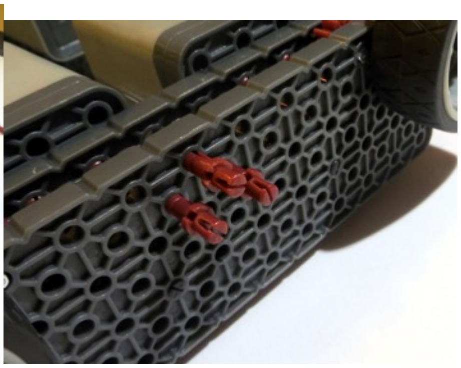
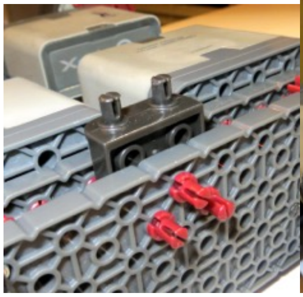
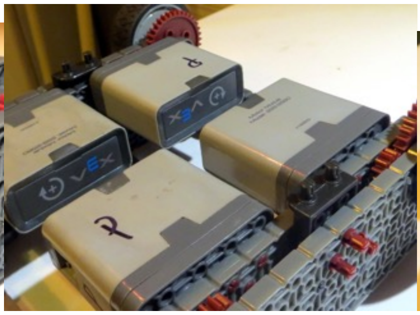
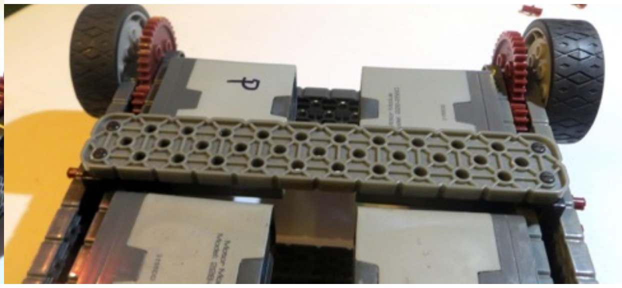


Connecting Left and Right Drivetrains

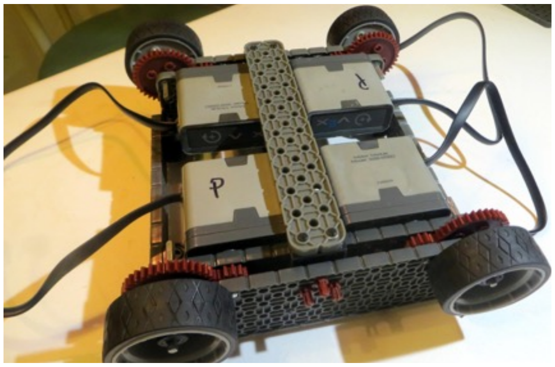
Step 1. Attach Top Cross supports.   Supplies (3) 2x10 beams



Step 2. Bottom support. Supplies (6) triple connectors, (2) 5x2 brackets, (1) 2x12 beam

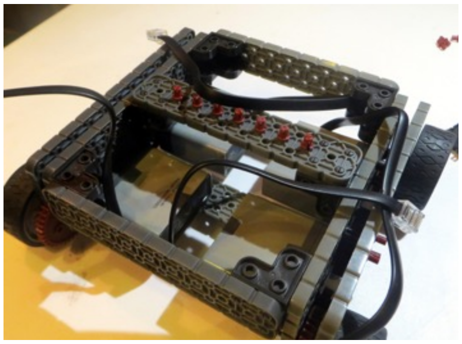







Step 2. Route smart cables

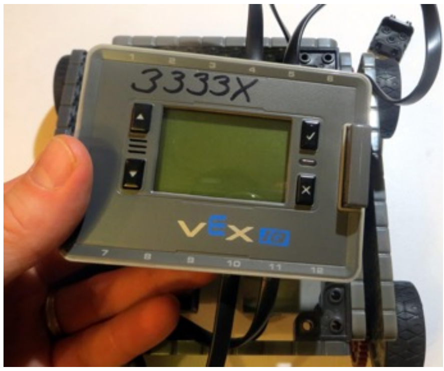


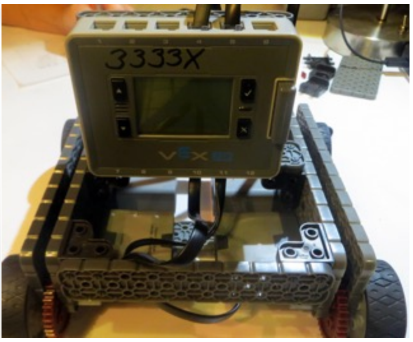
Hooking Up The Brain

Step 1. Supplies (6) normal connectors



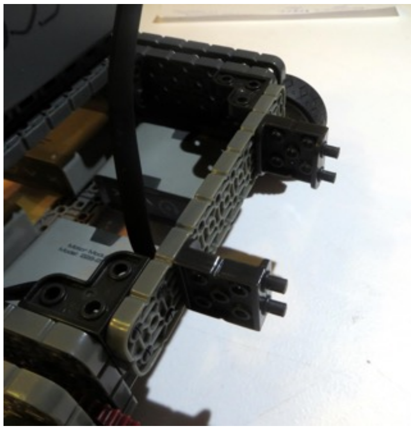
Step 2. Attach brain.

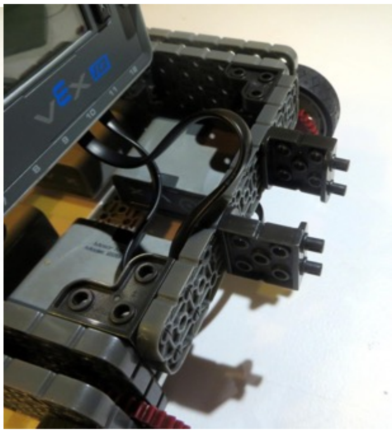


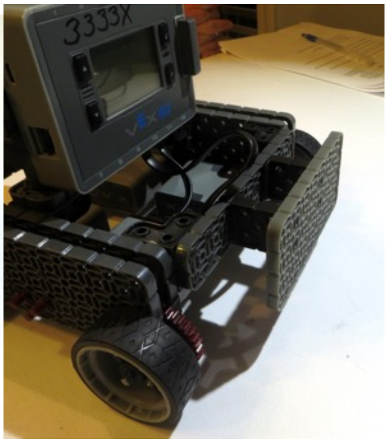


Adding Bumpers

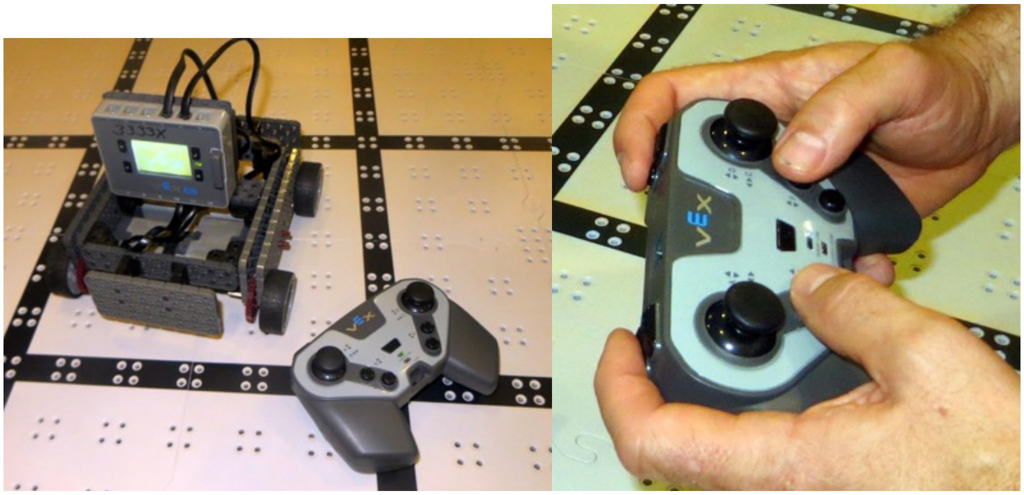
Supplies (4) 2x5x2 brackets, (2) 4x8 plates



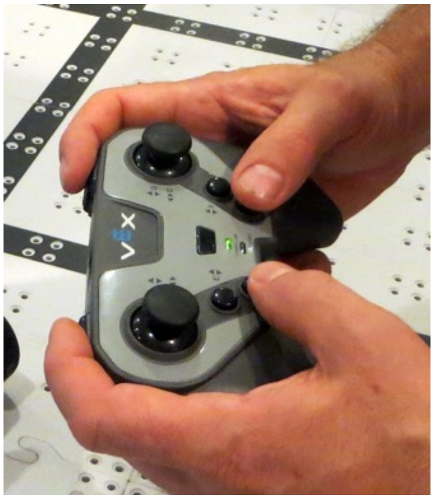




Running the Robot



Simultaneously press the upper E & F buttons and the lower R & L buttons to move forward



Simultaneously press the lower E & F buttons and the upper R & L buttons to move backwards

Note that with special programming the joysticks could be used to drive all 4 motors.

CODE - Cut & Paste everything 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);
    }
}
```