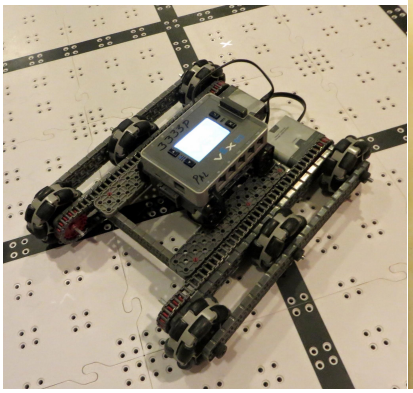
Six Omni Wheel Chain Drivetrain



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.

[Note, this can be built with only 4 omni wheel, just leave out the middle wheel on each side]

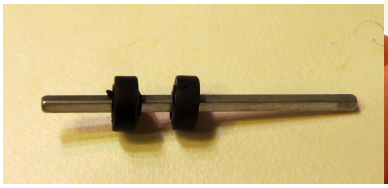
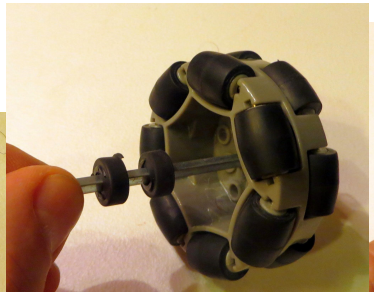
Brain & Battery	(4) medium-small sprockets	(4) 2x5 brackets	(6) 2x standoffs
Controller	About 4 ft of chain	(6) #3 shafts	(4) 2x20 beams
(2) motors	(6) thin washers	(20) rubber washers	(5) 2x12 beams
(6) Omni wheels	(20) spacer washers	(12) 2x3x2 corner brackets	handfull of single connectors
(2) smart cables			

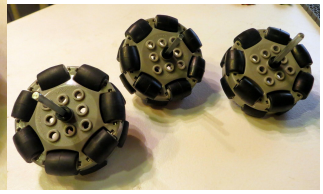
This drivetrain will consist of two sides that will be built separately and then attached together. If there are enough people helping with the build, then you could assign each side to a couple kids. The two sides are identical to each other.

One Side of Drivetrain

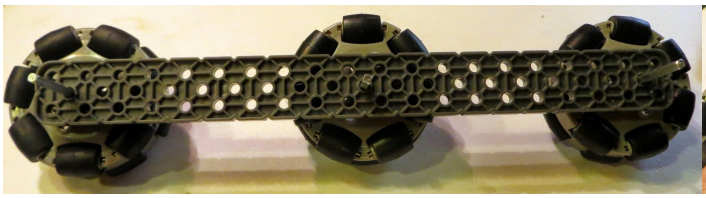
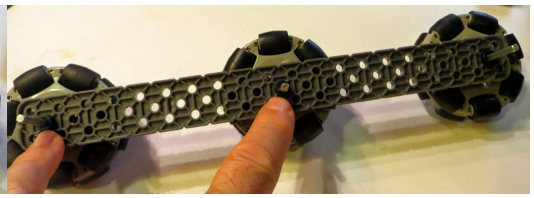
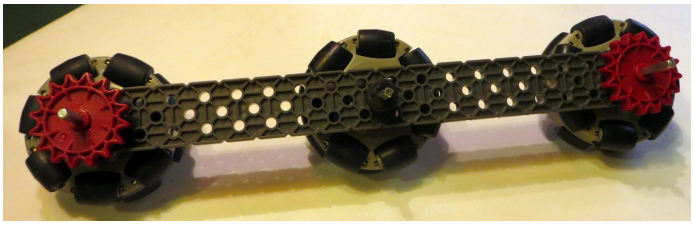
Build two complete sides using this same set of instructions

Step 1. Build wheels. Supplies: (3) omni wheels, (3) #3 shafts, (6) rubber washers, (3) spacer washers

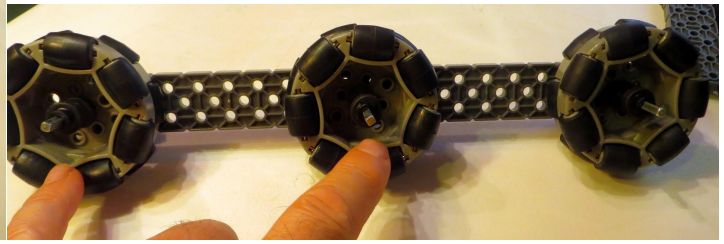





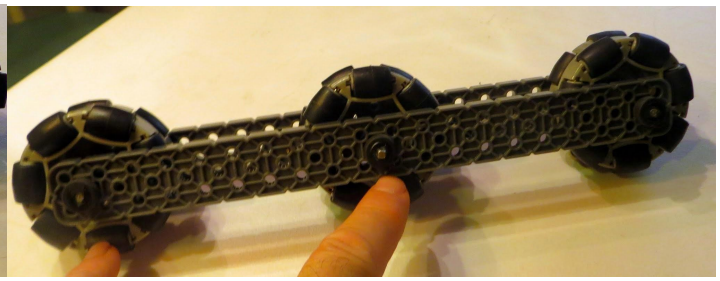
Step 2. Attach wheels & set sprockets. Supplies: (1) 2x20 beam, (2) medium-small sprocket, (3) spacer washers, (1) rubber washer

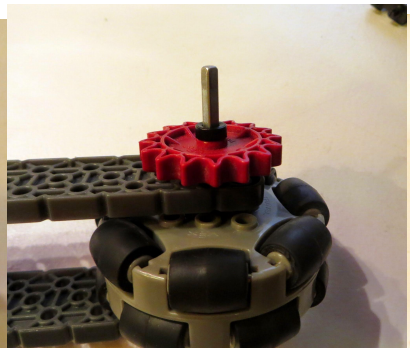
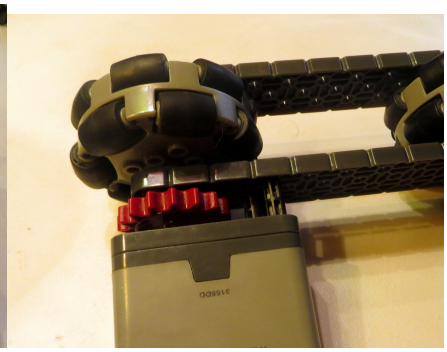
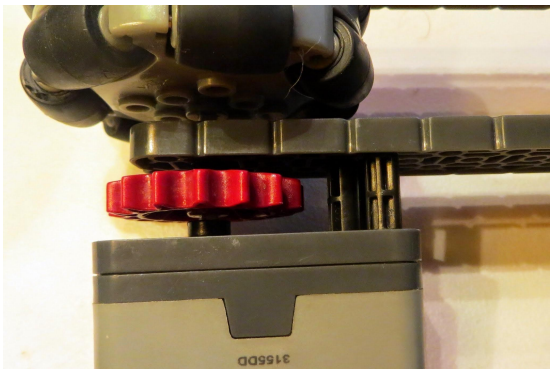
Step 3 Wheel sandwich. Supplies: (3) spacer washers, (3) thin washers, (1) 2x20 beam, (3) rubber washers

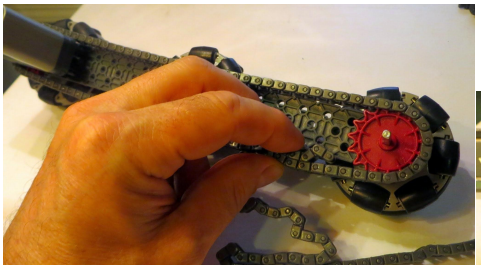
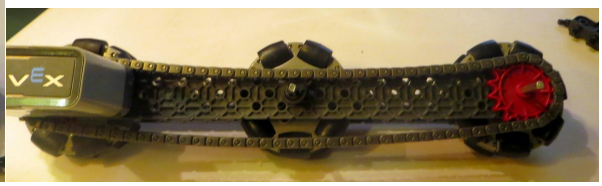
Add spacer washers, then attach beam, then thin washers, then rubber washers

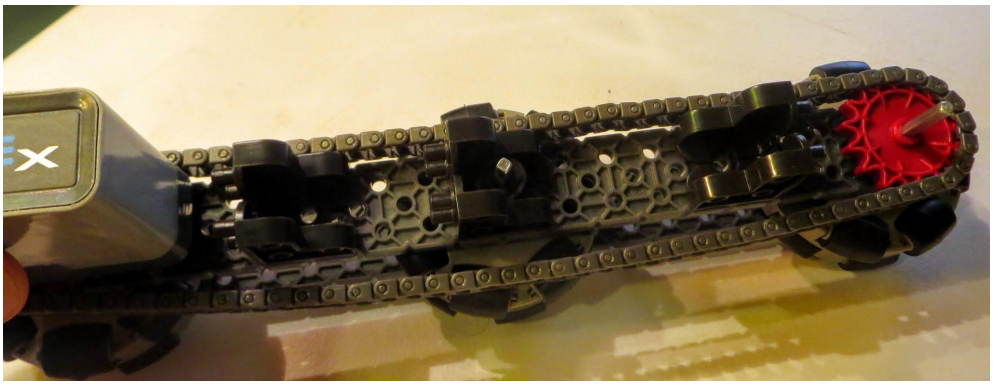
Step 5. Motor mount. (1) motor, (3) 2x standoffs

Step 6. Attach chain. Supplies: about 2 ft of chain

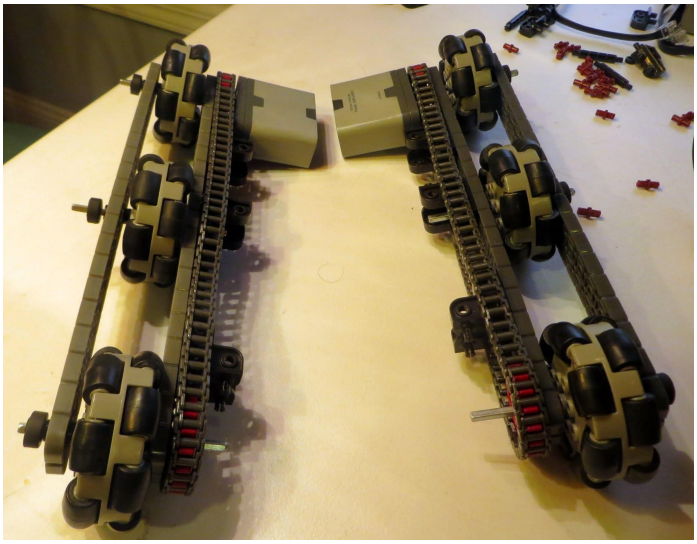
Step 7. Attach brackets. Supplies (6) 2x3x2 bracket connectors

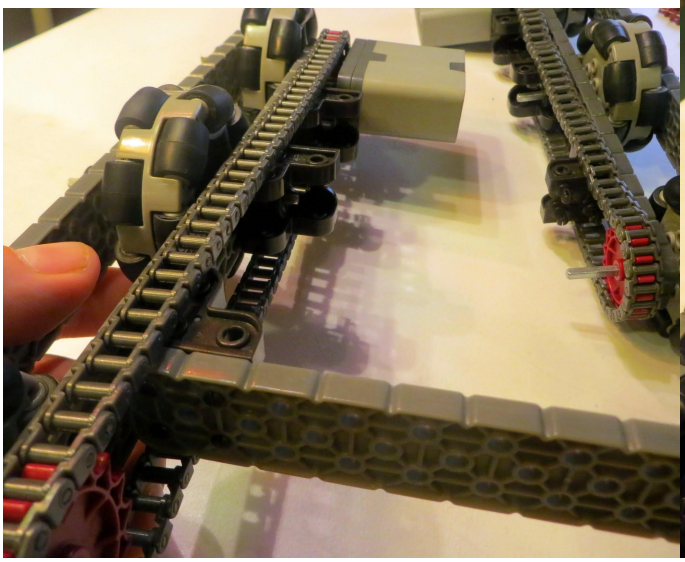


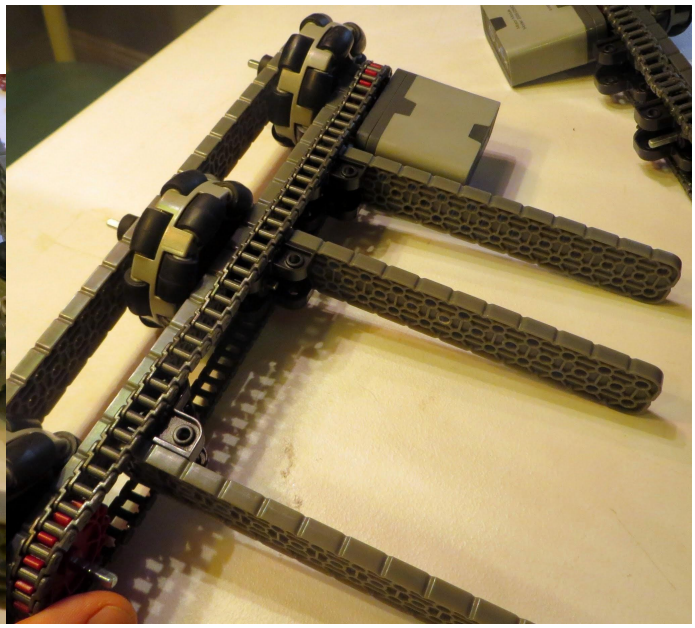
After the first side is built, build another one that is just the same.

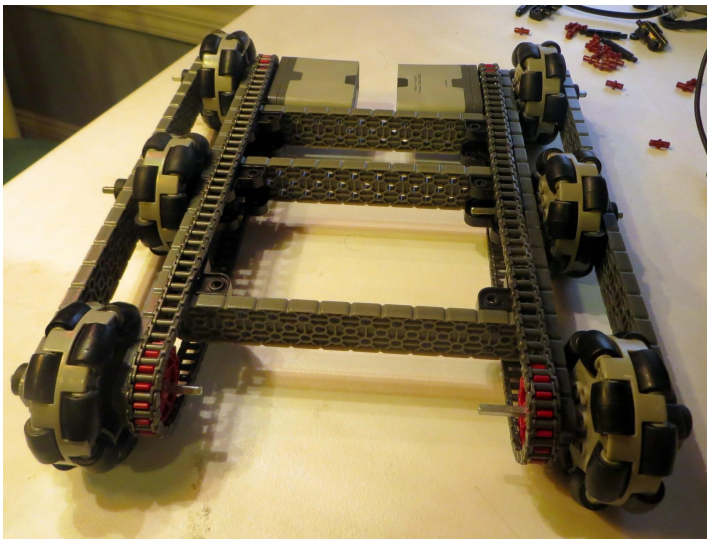
Connecting Left and Right Drivetrains

Step 1. Attach Cross supports. Supplies (3) 2x12 beams

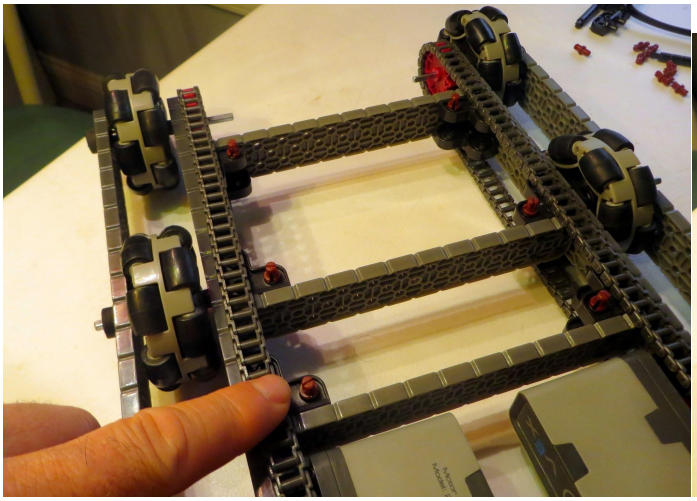


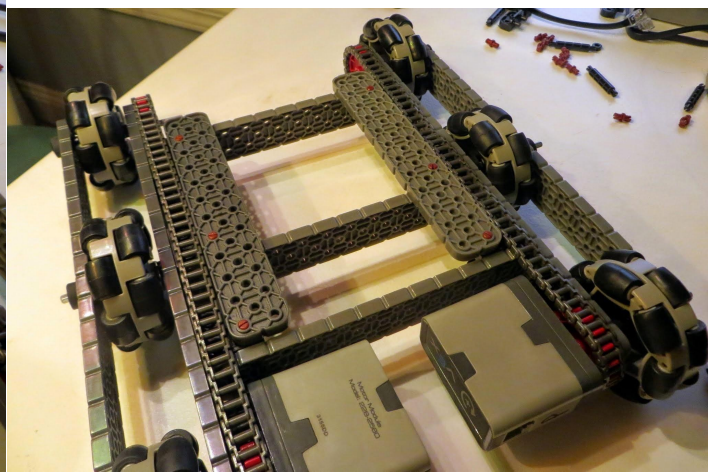






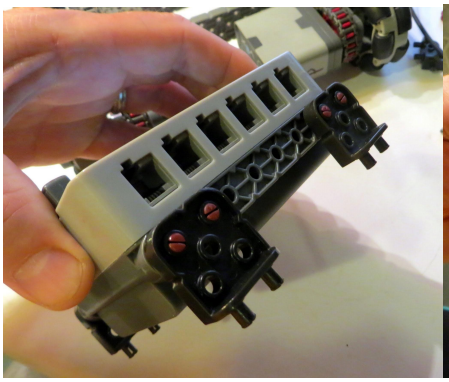
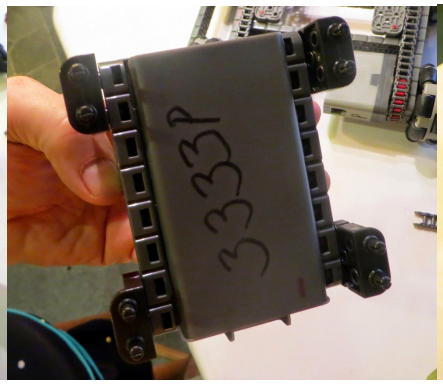
Step 2 Top supports. Supplies: (2) 2x12 beams, (6) normal connectors

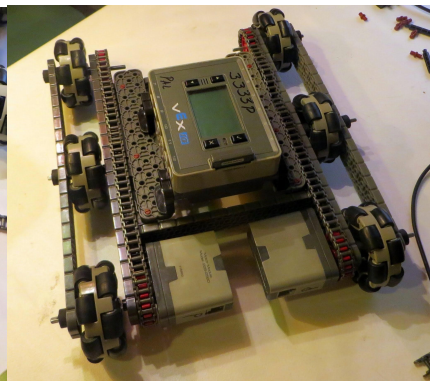




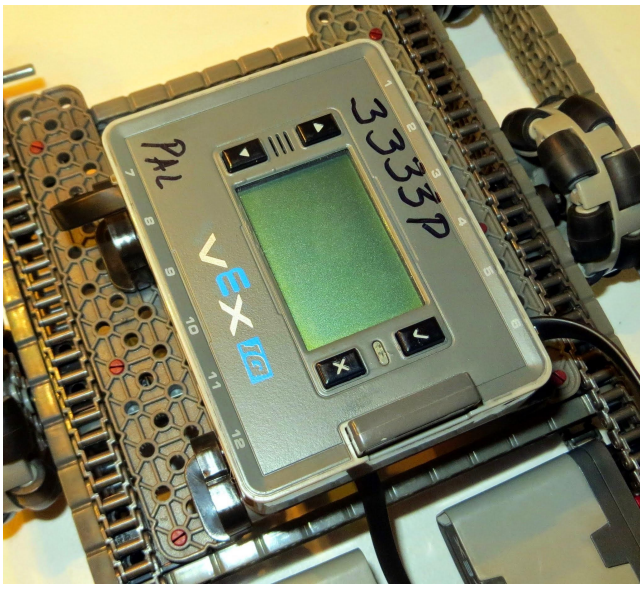
Hooking Up The Brain

Step 1. Supplies (4) 2x5 brackets, (8) connectors

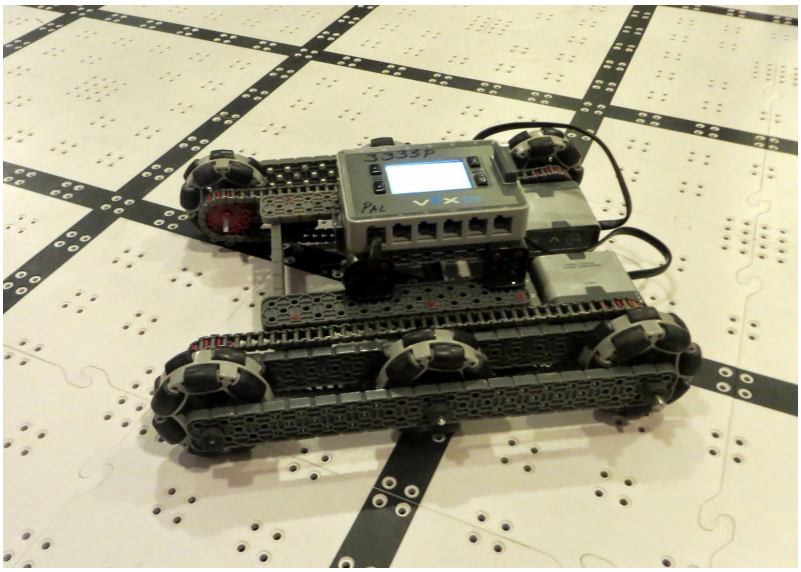


Step 2. Connect smart cables to ports 6 & 7. (2) smart cables



Running the Robot

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



Note that with special programming this drivetrain can run even better.

#CODE Copy & 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);
	}
}
```