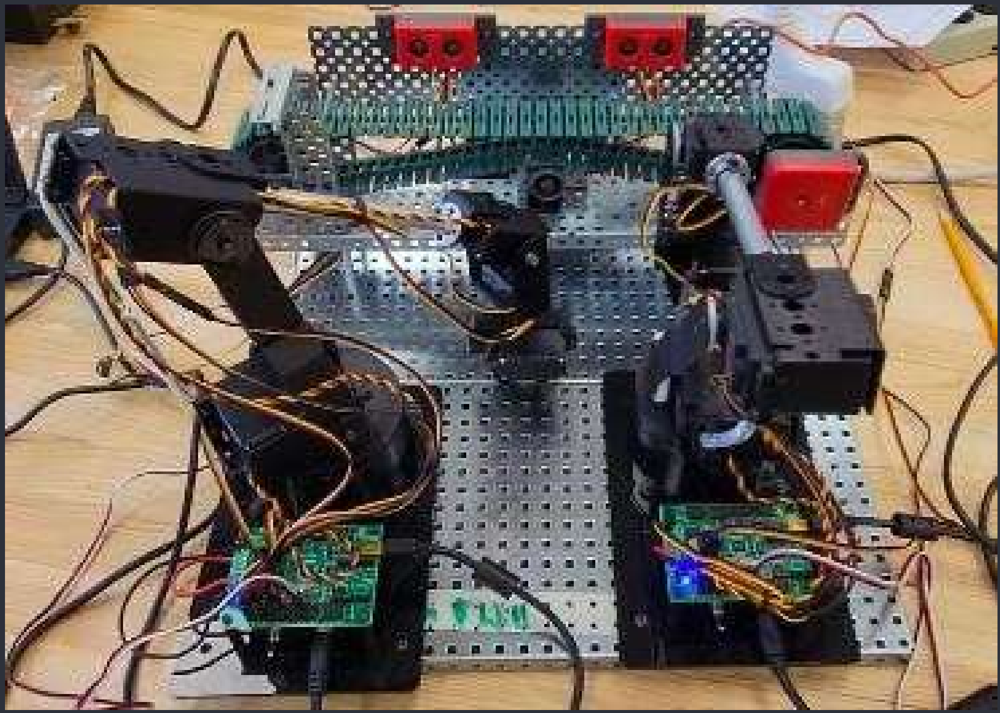
Blackstone Valley Tech

March 1st 2023

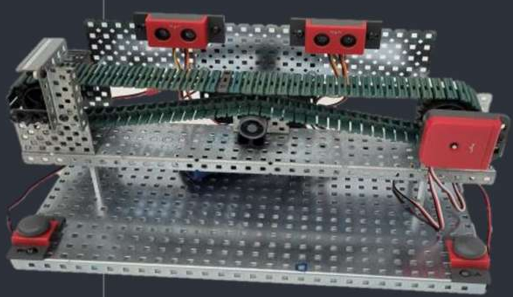
Transfer System Project

<Om Patel>

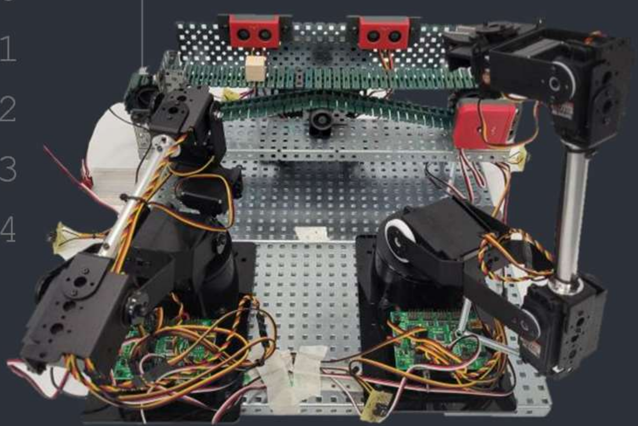


The Systems

Before



After



In this project one of the main components was to build a transfer system from scratch.

We used items such as optical shaft encoders, ultrasonic range finders and other materials from the vex catalog.

All our design choices were made in consideration of all parts of the project in including the coding and robot systems.

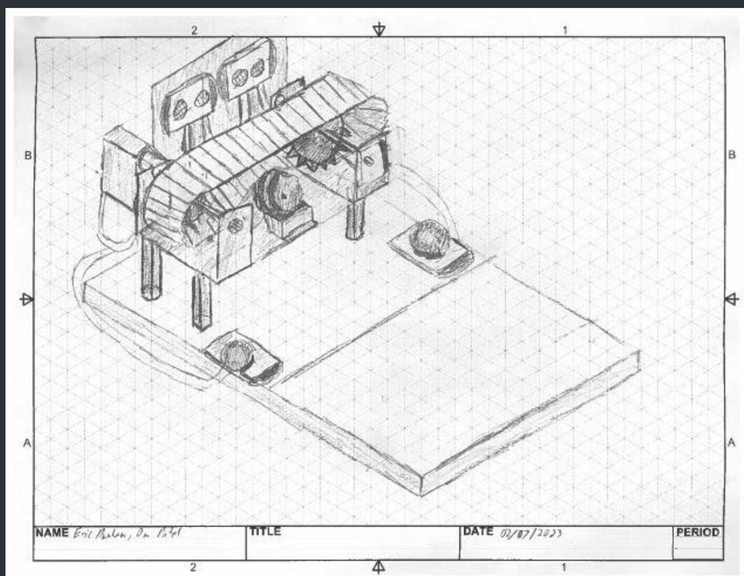
The top picture indicates our first ideas with out having our Lynx Motion Robots in place and the the one below is our final.

Blackstone Valley Tech

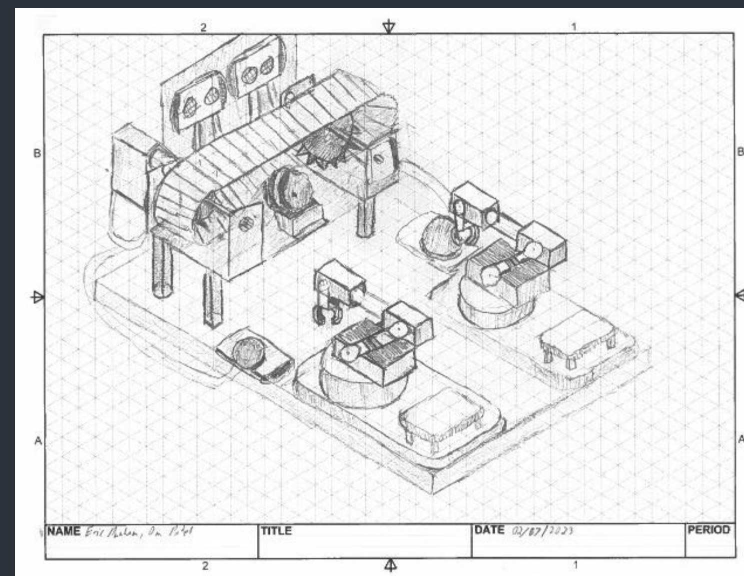
March 1st 2023

Sketches

Before



After



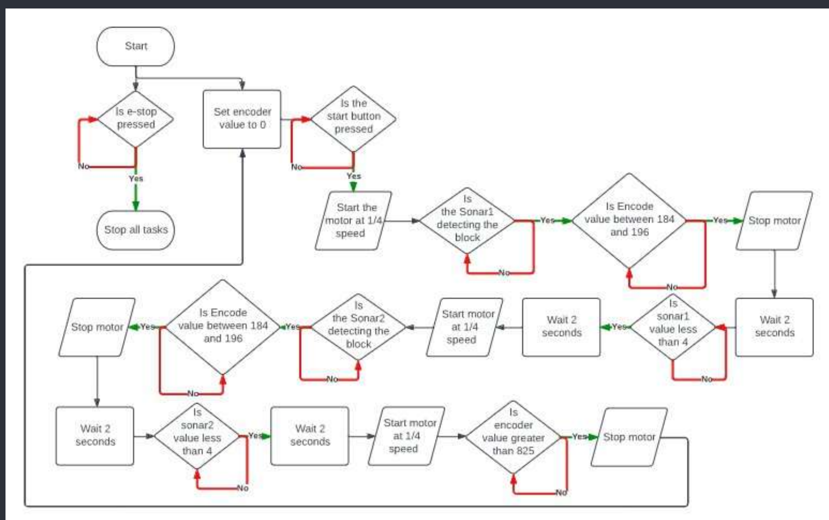
Our first design consisted of just our conveyor belt and the sensors need to stop and continue at two different locations with our second considering the locations of our two robots. This section of the project help me practice my brainstorming and sketching skills.

Blackstone Valley Tech

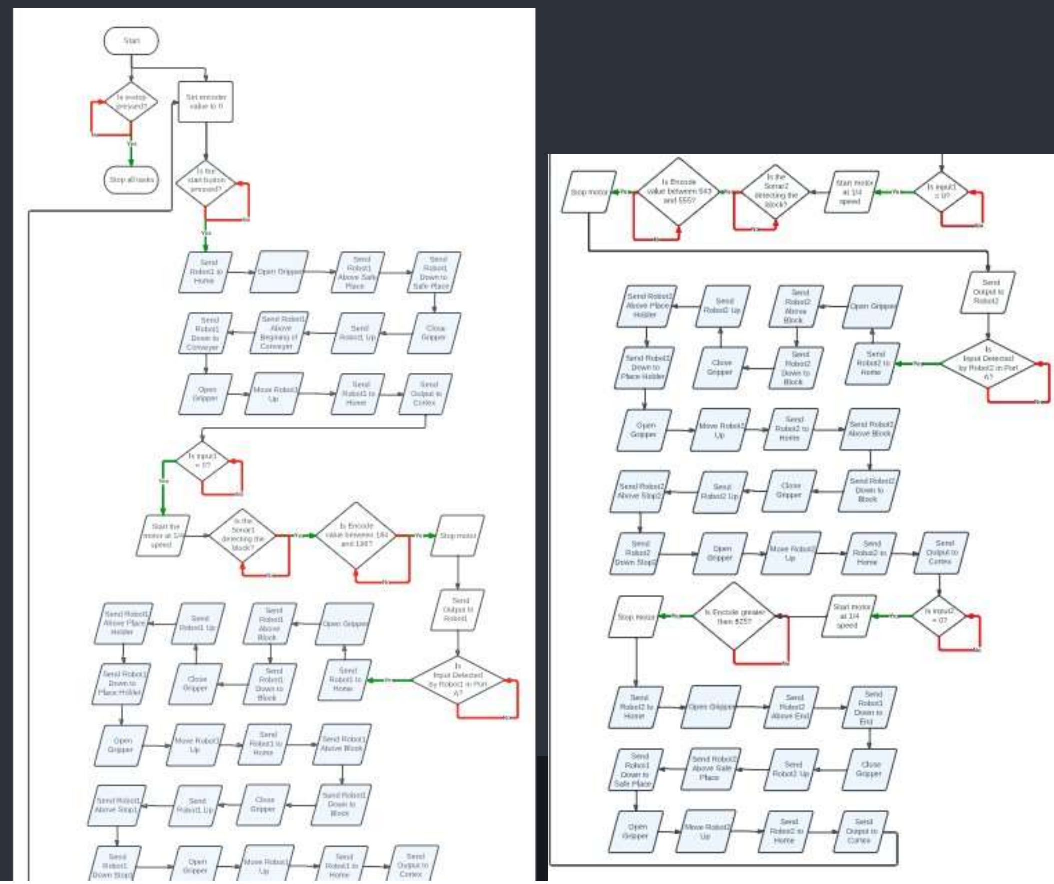
March 1st 2023

Flowcharts

Before



After



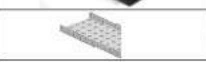
Flowcharts where essential to this project as they where an easy ways to show and document all of our ideas. It also shows all the processes our system goes through.

Blackstone Valley Tech

March 1st 2023

Cost Estimate

Our cost estimates allowed my partner and I to make sure our transfer system is cost effective and allows us to practice our time management skills.

Picture	Description	Cost Per Item	Number Used	Total Cost of Item
	**USB A-A Tether Cable, 6' (1.8m):** Cable for connecting VEXnet components to sync and run in tether mode	$ 10.00	1	$ 10.00
	**VEX Cortex Microcontroller:** The brain of every VEX robot. Coordinates the flow of information and power on the robot. All other electronic system components must interface to the Microcontroller.	$ 250.00	1	$ 250.00
	**7.2V Robot Battery NiMH 2000mAh:** Battery provides a rechargeable power source for use with your VEX robots.	$ 20.00	1	$ 20.00
	**Ultrasonic Range Finder:** Use this sensor to measure distance using high frequency sound waves. Sensor can measure in inches or centiemeters.	$ 30.00	2	$ 60.00
	**Optical Shaft Encoder:** The Quadrature Encoder can measure both the position and direction of rotation of a VEX shaft. This will allow you to calculate the speed of the shaft, as well as the distance it has traveled.	$ 10.00	1	$ 10.00
	**Bumper Switch:** Simple bumper switch. Ruggedized Bumpers allow the switch to be triggered by large impacts without sustaining damage.	$ 6.50	2	$ 13.00
	**2-Wire Motor 393:** Power motor. Add motors to power more wheels or add an end effector to take your robot to the next level.	$ 20.00	1	$ 20.00
	**Base Plate:** VEX Base Plates are designed to serve as a foundation for a variety of robotic applications. Use them to create a large stationary robotic arm, or interlock them to build larger manufacturing type systems. These pieces can also be used as large structural pieces.	$ 15.00	2	$ 30.00
	**Plate 5x25 holes:** Plate with holes on 0.500" increments and 1/8" diamonds in between. Can be cut on 0.500" increments.	$ 3.75	1	$ 3.75
	**C-Channel, 1x5x1x25 holes:** C-channel has holes on 0.500" increments. This structural member is perfect for building bigger robots. Excellent strength and twist resistance.	$ 5.00	1	$ 5.00
	**Drive Shaft 4":** The square shaft has rounded corners which allow it to spin easily in a round hole, while locking to a square hole.	$ 0.75	2	$ 1.50
	**Bearing Flat:** The Bearing Flat mounts directly on a piece of VEX structure and supports a shaft which runs perpendicular and directly through the structure.	$ 0.50	3	$ 1.50
	**Standoff, 1" Long:** A standoff is used in mechanics and electronics to separate two parts from one another. All VEX Standoffs have 8-32 threads.	$ 0.40	2	$ 0.80
	**Standoff, 2" Long:** A standoff is used in mechanics and electronics to separate two parts from one another. All VEX Standoffs have 8-32 threads.	$ 0.80	4	$ 3.20
	**Shaft Spacer, Thin (4.6mm):** A spacer is used to create a space between two objects, often to properly position them.	$ 0.15	3	$ 0.45
	**Shaft Spacer, Thick (8mm):** A spacer is used to create a space between two objects, often to properly position them.	$ 0.15	5	$ 0.75

Picture	Description	Cost Per Item	Number Used	Total Cost of Item
	**Screw, 8-32 x1/2" Long:** 8-32 Screw with Button Head. Standard 3/32" VEX Allen Drive. High Strength Stainless Steel.	$ 0.10	32	$ 3.20
	**Screw, 8-32 x3/4" Long:** 8-32 Screw with Button Head. Standard 3/32" VEX Allen Drive. High Strength Stainless Steel.	$ 0.10		
	**Screw, 8-32 x1" Long:** 8-32 Screw with Button Head. Standard 3/32" VEX Allen Drive. High Strength Stainless Steel.	$ 0.15	2	$ 0.30
	**Nuts 8-32 Keps:** Keps Nuts have integral external tooth lock washer,	$ 0.03	22	$ 0.66
	**Chain Attachment Links:** Price based per link. Allows material to be attahed directly to the chain links.	$ 0.50	1	$ 0.50
	**Single Bogie Wheel Assemblies:** For use with the Tread links.	$ 1.50	1	$ 1.50
	**Tread Link:** Price based on 85 pieces per track. Use this tank tread to build robot tracks which can overcome tough terrain, or build a conveyor belt for scooping up objects.	$ 5.00	85	$ 425.00
	**Tank Tread Drive/Idler Wheels:**	$ 2.00	2	$ 4.00
	Cable, PWM Extension, 6": Extension cables are used to extend the length of a 3-wire cable such that a motor/servo or sensor can be farther from the VEX Microcontroller.	$ 5.00	4	$ 20.00
			Total	$ 885.11

Team Members		Time Employed (Minutes)	Employee Rate	Cost of Employees	Total Cost of Employees	Total Cost of Parts	Total Cost
Team member 1	Eric Dunham	300	$ 15.00	$ 4,500.00	$ 9,000.00	$ 885.11	$ 9,885.11
Team member 2	Om Patel	300	$ 15.00	$ 4,500.00			
Team member 3			$ 15.00				
Team member 4			$ 15.00				

Blackstone Valley Tech

March 1st 2023

Programming

Before

This project tested our programming skills having to use all the programming knowledge we know.

In the first part we used an ultra sonic sensor to detect when a block is placed back down and off of the conveyor belt and a encoder to make sure it stops in the exact spot every time.

```
Instructions:
1) Create and start e_stop
2) Start program when the start button is pressed
3) Run the conveyer until detected by sonar and between encoder values in two di
4) Detects when block is placed and run the motors again
5) When block reaches end of conveyer allow for the program to run again when st

*/
task e_stop()
{
  while(true) //run forever
  {
    if(SensorValue(estop) == 1) // is the estop button pressed
    {
      stopAllTasks();       // ends the program and all tasks including task mair
    }
    wait1Msec(10);    // prevents the current task from using majority of availa
  }
}
task main()
{
  SensorValue(encoder) = 0; // sets encoder to zero
  startTask(e_stop); // allows the estop to work no matter where in code
  while (true) // run forever
  {
    if (SensorValue(startBut) ==1) // run commands when start button is pressed
    {
      SensorValue(encoder) = 0; // sets encoder value to zero
      startMotor(leftMotor, 32); // runs motor 1/4th speed until sonar1 detects
      waitUntil(SensorValue (sonar1) <4 && SensorValue(encoder) <196 && SensorVa
      stopMotor(leftMotor); // stops motor
      wait(2); // wait 2 seconds for arm to pick up block
      waitUntil (SensorValue(sonar1) <4); // wait for sonar1 to detect the block
      wait(2); // wait 2 seconds for arm to move away from the block
      startMotor (leftMotor, 32); // runs motor 1/4th speed until sonar2 detects
      waitUntil(SensorValue (sonar2) <4 && SensorValue(encoder) <555 && SensorVa
      stopMotor(leftMotor); // stops motor
      wait(2); // wait 2 seconds for arm to pick up block
      waitUntil (SensorValue(sonar2) <4); // wait for sonar2 to detect the block
      wait(2); // wait 2 seconds for arm to move away from the block
      startMotor (leftMotor, 32); // start motor until the encoder is greater th
      waitUntil (SensorValue(encoder) >825);
      stopMotor(leftMotor); //stop motor
    }
  }
}
```

Programming

After

In the second part we made sure to keep the emergency stop from the first code and the main concept. Some things we changed when the motors start and what happens when they stop. To start they must wait until they receive an input from a robot and when they stop they must send an output to a robot. We also had to make sure the conveyor doesn't let the block roll off the belt in this version.

```
Instructions:
1) Create and start e_stop
2) Once Robot 1 is in a safe position the system will start (signaled from Robot
3) Run the conveyer until detected by sonar1 and between encoder values
4) Send an output back to robot1
5) Once Robot 1 is in a safe position the system will continue (signaled from Rc
6) Run the conveyer until detected by sonar2 and between encoder values
7) Send an output to robot2
8) Once Robot 2 is in a safe position the system will continue (signaled from Rc
9) Run the conveyer until between encoder values
10) Send an output to robot2
11) When block reaches end of to home position allow for the program to run agai

*/
task e_stop()
{
  while(true) //run forever
  {
    if(SensorValue(estop) == 1) // is the estop button pressed
    {
      stopAllTasks();      // ends the program and all tasks including task mair
    }
    wait1Msec(10);    // prevents the current task from using majority of availa
  }
}
task main()
{
  SensorValue(encoder) = 0; // sets encoder to zero
  startTask(e_stop); // allows the estop to work no matter where in code
  while (true) // run forever
  {
    if (SensorValue(startBut) ==1) // run commands when start button is pressed
    {
      SensorValue(encoder) = 0; // sets encoder value to zero
      startMotor(leftMotor, 32); // runs motor 1/4th speed until sonar1 detects
      waitUntil(SensorValue (sonar1) <4 && SensorValue(encoder) <196 && SensorVa
      stopMotor(leftMotor); // stops motor
      wait(2); // wait 2 seconds for arm to pick up block
      waitUntil (SensorValue(sonar1) <4); // wait for sonar1 to detect the block
      wait(2); // wait 2 seconds for arm to move away from the block
      startMotor (leftMotor, 32); // runs motor 1/4th speed until sonar2 detects
      waitUntil(SensorValue (sonar2) <4 && SensorValue(encoder) <555 && SensorVa
      stopMotor(leftMotor); // stops motor
      wait(2); // wait 2 seconds for arm to pick up block
      waitUntil (SensorValue(sonar2) <4); // wait for sonar2 to detect the block
      wait(2); // wait 2 seconds for arm to move away from the block
     startMotor (leftMotor, 32); // start motor until the encoder is greater th
     waitUntil (SensorValue(encoder) >825);
     stopMotor(leftMotor); //stop motor
    }
  }
}
```

Blackstone Valley Tech

March 1st 2023

The System {In Progress}