STRATEGY & INNOVATION

DESIGN PROCESS

- In designing the base robot, we establish criteria at the outset (such as desired footprint, height, balance, drivetrain, sensors to be used, user interface, etc.) to arrive at a working prototype.
- Before creating the base robot we prepare some high level rudimentary drawings on a flip chart which we use to stimulate discussion amongst team members.
- We then construct an initial prototype of the robot starting from the motors and working outwards until we get to the wheels, frame, microcontroller positioning and finally the wiring.

MISSION STRATEGY

- Our mission strategy, anticipated score and expected number of points per second are clearly set out in the following diagram and table:

Zone 1: Collects Core Samples – 10 Seconds

Zone 2: Completes Escape Velocity, pushes Observatory's pointer to orange area, pushes Food Production bar to scoring position while placing Water Core Sample on top, collects Cone Module and Gerhard – 33 Seconds

Zone 3: Rolls all 3 Payloads down Space Travel Mission Model, pushes Solar Panel Array outwards, hits down Crater Crossing gate – 13 Seconds

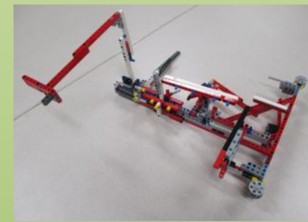
Zone 4: Inserts Tube Module, places Gerhard into Airlock Chamber – 8 Seconds

Zone 5: Activates 3D Printer, Completes Strength Bar, pushes other Solar Panel Array outwards, delivers 3 Satellites into the scoring area – 33 Seconds

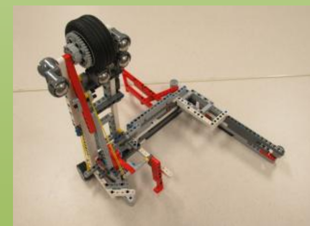
Analysis of Points Scored Per Second

Zone	Missions	Points	Time in Base Before Mission (s)	Robot Running Time (s)	Total Mission Time (s)	Clock at End of Mission	Points per Second
Zone 1	Extraction	26	0	10	10	0min 10sec	2.60
Zone 2	Escape Velocity, Observatory, Food Production with Water, Cone Module	84	11	33	44	0min 54sec	1.91
Zone 3	Space Travel, Solar Panel Array (ours), Crater Crossing (Gate)	64	12	13	25	1min 19sec	2.56
Zone 4	Tube Module, Space Walk Emergency	38	9	8	17	1min 36sec	2.24
Zone 5	Satellite Orbits, Strength Bar, 3D Printer, Crater Crossing (Cross), Solar Panel Array (theirs)	100	13	33	46	2min 22sec	2.17
Total	**Space Travel, Solar Panel Array, Extraction, Space Station Modules, Escape Velocity, Observatory, Satellite Orbits, Food Production**	**312**	**45**	**97**	**142**	**2min 22sec**	**2.3**

Our Solar Carts attachment pushes the solar panel forward, pushes the first payload down the ramp, and gets the other two payloads down the ramp.



Our Tela-Velocity attachment shoots the shuttle up the vertical post, and pushes the telescope arm into the orange region.



We use a pneumatic powered arm to lift the Strength Bar upwards. As the bar is lifted the attachment is released from the robot.

Bayview Glen School
275 Duncan Mill Road
Don Mills, ON
M3B 3H9

BAYVIEW GLEN
Whole Child. Whole Life. Whole World.

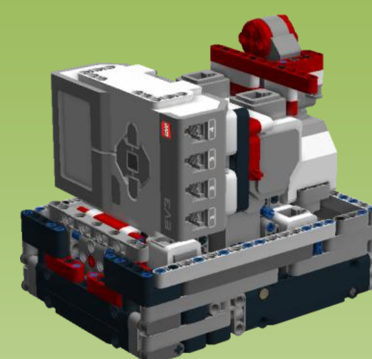
Texas Instruments
Harborn Dental
Qualcomm
Liberty development
FIRST Canada

Robot

TEAM 5831

CTRL Z

http://fllctrlz.github.io/into_orbit/



MECHANICAL DESIGN

DURABILITY

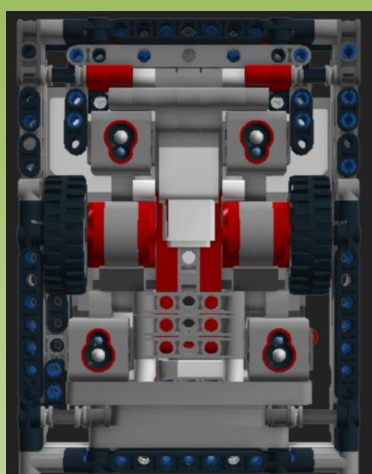
- We strive to design and build a base robot which is robust, and features a frame around the chassis which provides structural rigidity as well as a means to square up against solid objects, like mission models.

MECHANICAL EFFICIENCY

- Most of our robot attachments are designed to simply “drop in” so that they can be installed in seconds without the need to fumble with friction pegs when in base.
- We created a compact robot and few multi-purpose attachments to efficiently use our scarce LEGO parts.

MECHANIZATION (PROPULSION AND EXECUTION)

- We use small, narrow and hard wheels to increase accuracy in both straight movements and turns.
- When accelerating from a stationary position, our robot uses a Ctrl-Z-designed ramp up feature to minimize the likelihood of a wheel slipping. This provides for greater repeatability of the robot’s movements.



PROGRAMMING

QUALITY

- We have created our own Move block that corrects error (to keep the robot straight). It also has an acceleration and deceleration feature. This improves repeatability.
- We have created a selection program (see center image on right) which allows us to quickly and efficiently choose missions (or even re-run missions quickly) during matches. Audible sounds and easy-to-understand EV3 screen messages allow the user to select programs with ease.
- Our “Check-Abort” MyBlock constantly checks whether the EV3 up arrow button is pressed. We can interrupt our missions at any time and still stay within the overall selection program to avoid the need to exit the program and find the next program or mission.

EFFICIENCY

- We use subroutines (i.e. MyBlocks) with standardized exit condition parameters to reuse code whenever possible and to keep our program sizes manageable.
- Our MyBlocks, variables and file access blocks are given descriptive and meaningful names to allow us to easily understand what our programs are doing.
- We annotate our work to help us debug programs and to better understand each programming block’s purpose.

AUTOMATION / NAVIGATION

- We program our robot to drive along walls and square to mission models, walls, and lines to maximize repeatability and minimize the likelihood that our team will incur a touch penalty.
- We program the robot to use 4 light sensors with custom calibration for navigational flexibility and rotation sensors to measure distance.
- We follow lines using a Proportional Integral Derivative algorithm and we make frequent use of our square up subroutine which allows the robot to position itself perpendicularly to any line (from virtually any angle of attack).

PROGRAMMING (CONTINUED)