

Identify the need

- The situation: Our autonomous was ineffective
- Challenge statement: Create a ^{dynamic} program/thing that can track a robot's location and direct it on a $12 \times 12 \text{ ft}^2$ FTC game field

Describe

Describe the need

- Can direct a robot to any arbitrary point
- Existing approaches: Time, accelerometer, light

Constraints

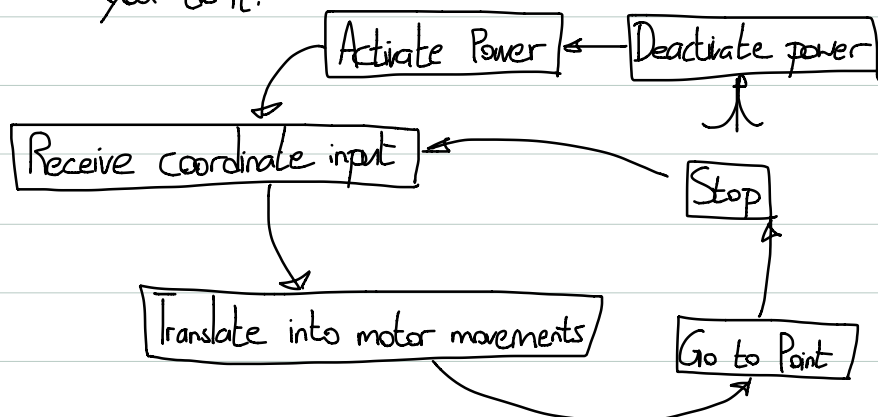
- Must use FTC allowed parts/materials only

Requirements

- Must be fast enough to complete an autonomous routine within 30 seconds
- Can withstand the occasional violent blow

Characterize and Analyze the system

- Produce a functional model - I didn't make ^{one} so, how would you do it?



Quantify constraints, requirements

Constraints

- See game manual

Requirements

- The robot cannot stop for more than sec between each movement and must travel at speed \geq cm/s
- Can withstand \geq N of force for \geq s

Divide project into subsystems

- Hardware subsystem
- Software subsystem

Understand equipment available (Tetrix/Robot)

- Encoders
- Axle stability
- RobotC
- Trigonometry

Set up demos

Generate

"Generated" Concepts - I drifted between ^{research,} generate, and embody

- ① Accelerometer and Compass
- ② Encoder and Compass
- ③ Encoder and $\frac{\text{Arc}}{\text{radius}} = \theta$
- ④ Encoder, $\frac{\text{Arc}}{\text{radius}} = \theta$, and absolute position

Concept Selection - my process was try the first thing that came up

How would you do it knowing some strategies?

Example Pugh chart:

1-5 rating \times %

	5 =	Weight	①	②	③	④
Physical Complexity	Not complex	10%	1	4	3	2
Physical robustness	practically indestructible	10%	3	2	2	2
Physical bulk	$\leq \frac{18^3 \text{ in}^3}{8^3}$	15%	5	4	4	4
Software Complexity	Easily understandable	15%	1	2	3	3
Reliability	100% reliability	25%	1	1	3	3
Satisfies all con/reqs	Goes above and beyond	25%	1	1	4	5
Total		100%				

