BB-8

By
Amar Hamdan Thayyil - 132001007
Labeeb Nassar - 132001019

Problem Statement

Create a replica of BB-8 Droid from the movie StarWars

(A fully autonomous spherical robot)

Project Guide

Dr. Anand TNC

Inspiration

Inspiration

BB-8 is a droid character in the Star Wars franchise. It's a spherical robot with a free-moving domed head, BB-8 droid is portrayed by both a rod puppet and a remote-controlled robotic unit.

Introduction

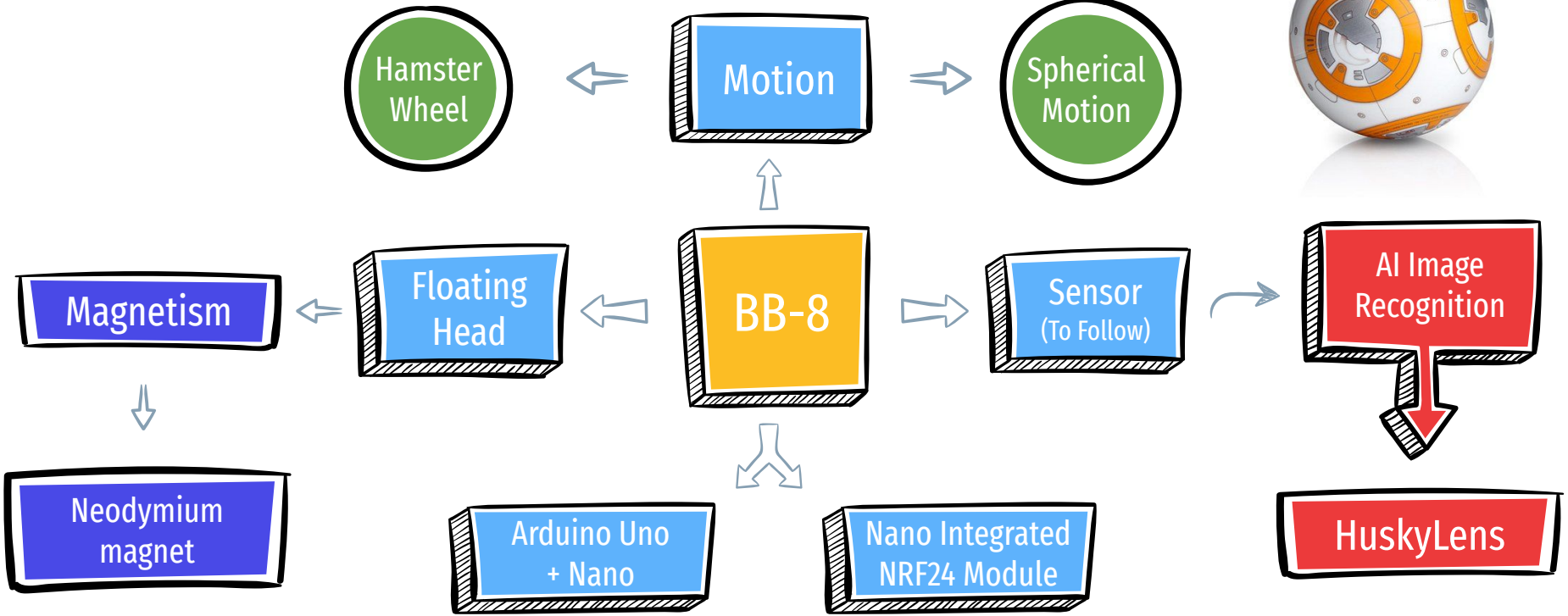
Introduction

BB-8 is a spherical robot with a dome shaped head. The eye catching feature of the robot is the dome shaped head that always stays upright over the spherical body that moves in all directions.

This project focuses on the realization of this droid character. The will be fully autonomous and is built and programmed to follow a person or an object that is moving around. This feature is achieved by an AI image recognition board with a built-in camera.

Plan of Action

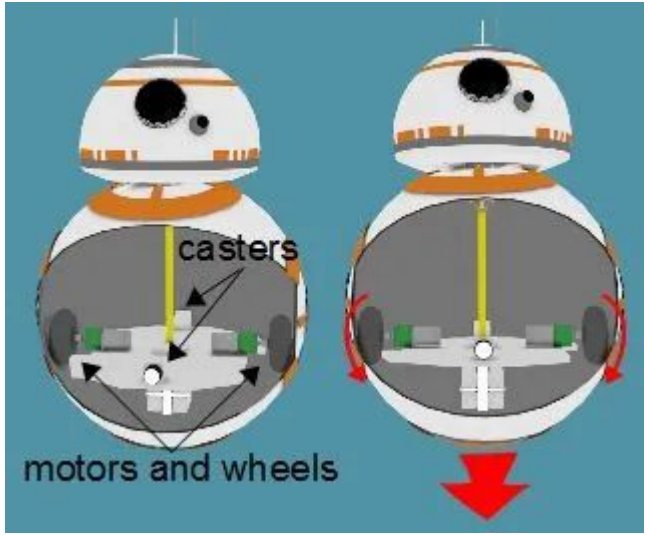
Plan Of Action



Design and Components

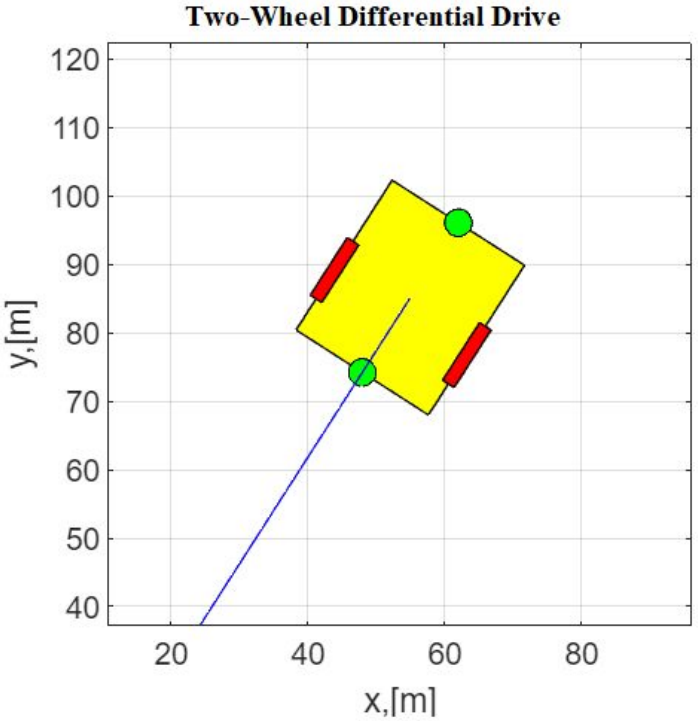
Hamster Wheel Mechanism

- The hamster drive mechanism consists of a 2-wheeled vehicle that sits in the spherical body
- It not attached to the spherical body's inner surface. Further two castors are used to stabilise the 2-wheel drive inside the body.
- The robot moves when the wheels move in the same direction and the robot turns when the wheels move in the opposite direction relative to each other, hence the drive is differential as each wheel has its own motor.



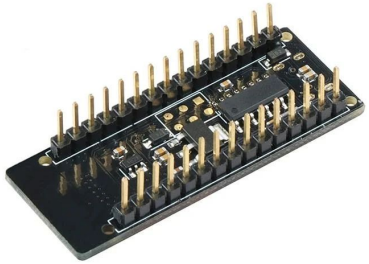
WHEEL CONFIGURATION

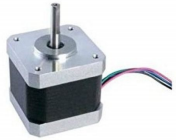
- The internal vehicle of the BB-8 robot has a two-wheeled differential drive configuration.
- With two castor wheels at the mid-ends of the vehicle base
- The entire robot is non-holonomic

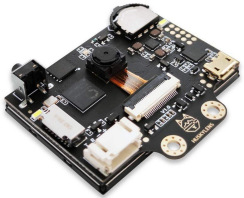


Components

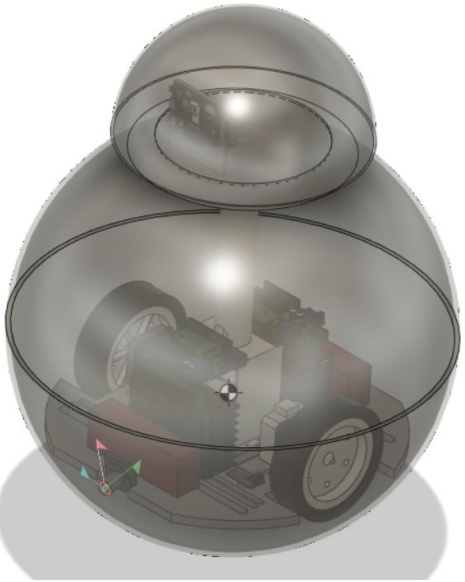
- RF Nano Integrated wifi module
- Nema 17 4.8Kg-cm
- TB6600 Stepper motor driver
- Neodymium (25mmx5mm)
- Ball Castor wheel
- Wheel 100mm
- HC-sr04 Ultrasonic sensor
- LIPO Battery 11.1V
- Arduino UNO
- HuskyLens AI Camera
- Zip Ties

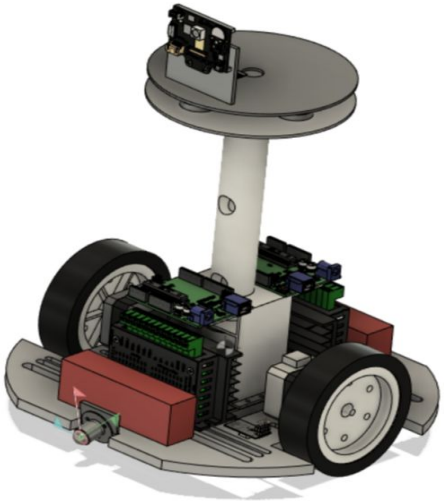




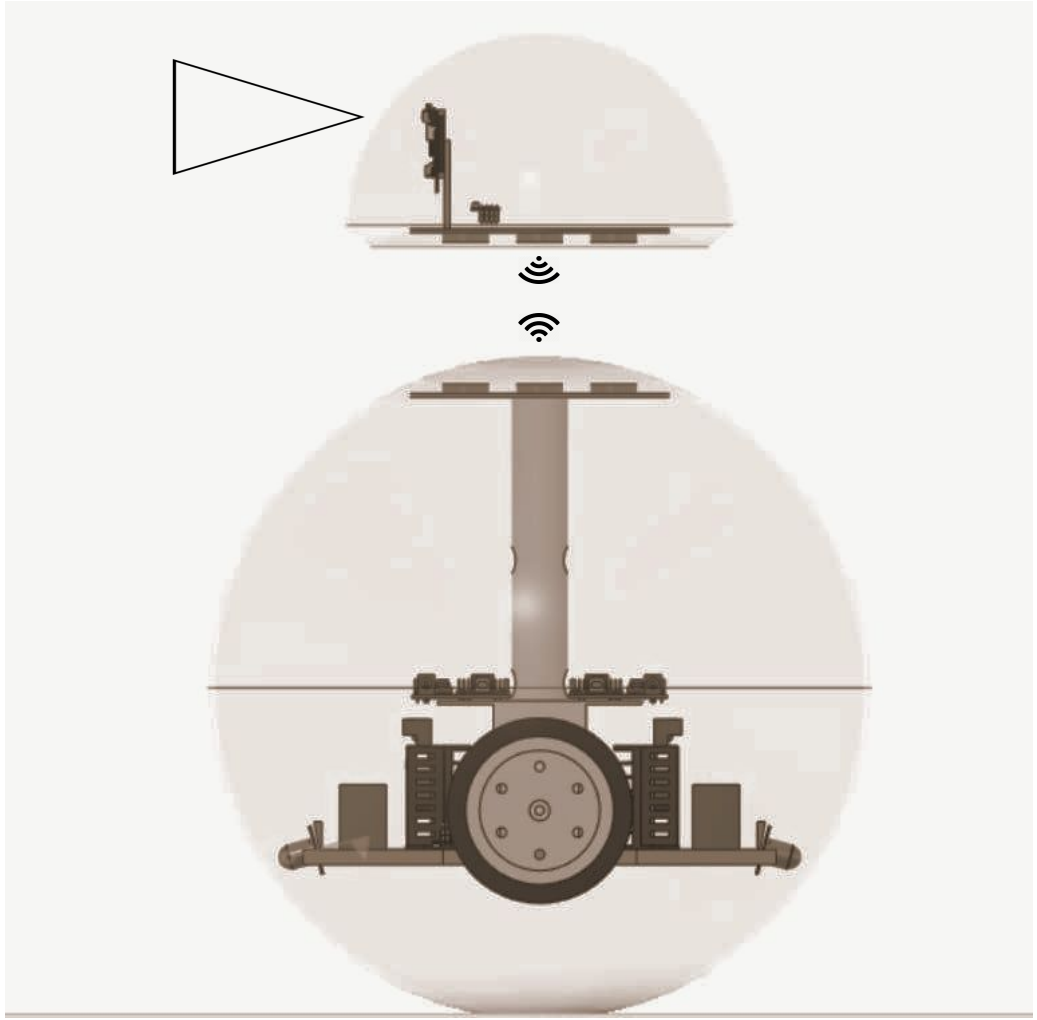


CAD MODEL (Basic Design)





The Algorithm



Microcontroller - RF Nano Integrated NRF24L01 Wireless Module.

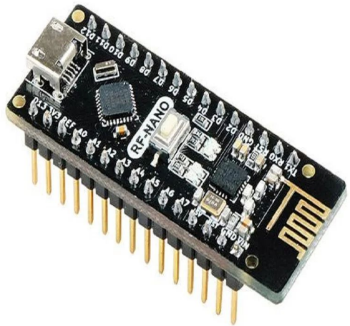
RF Nano is an Arduino compatible board with an NRF24L01 wifi module integrated into it.

The board in run with an ATMEGA328P chipset.

This wifi module uses the 2.4 GHz band and it can operate from 250 kbps up to 2 Mbps.

2 modules can easily communicate within a 10m range, with an open space range upto 100m.

The board is programmed using the Arduino IDE with embedded C programming.



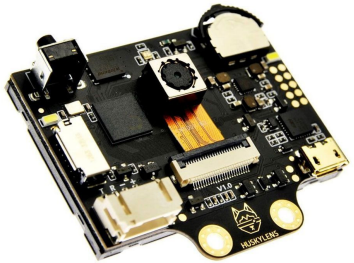
HuskyLens

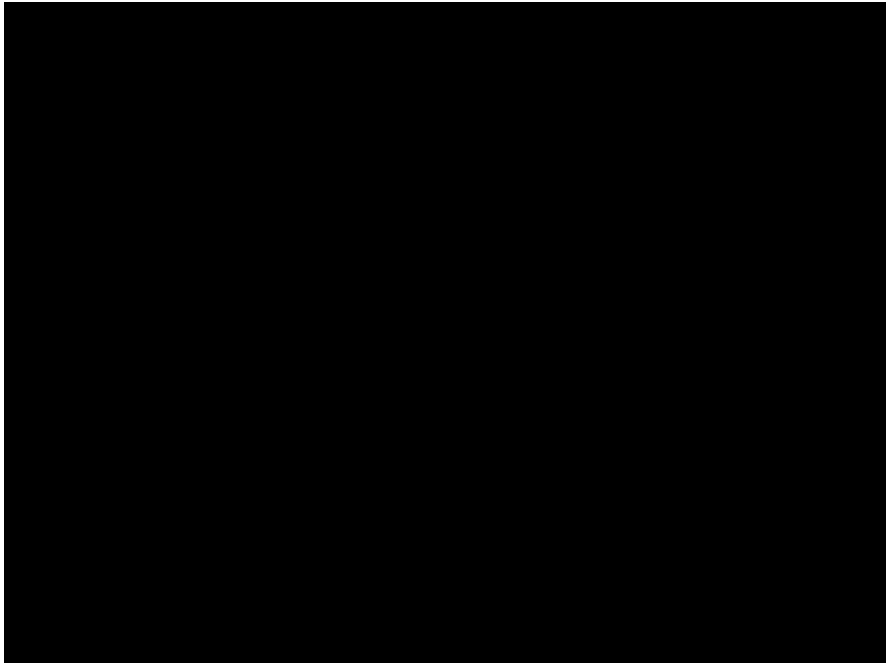
HuskyLens is an easy-to-use AI machine vision sensor. It can learn to detect objects, faces, lines, colors and tags.

Through the UART / I2C port, HuskyLens can connect to Arduino, Raspberry and other microcontrollers.

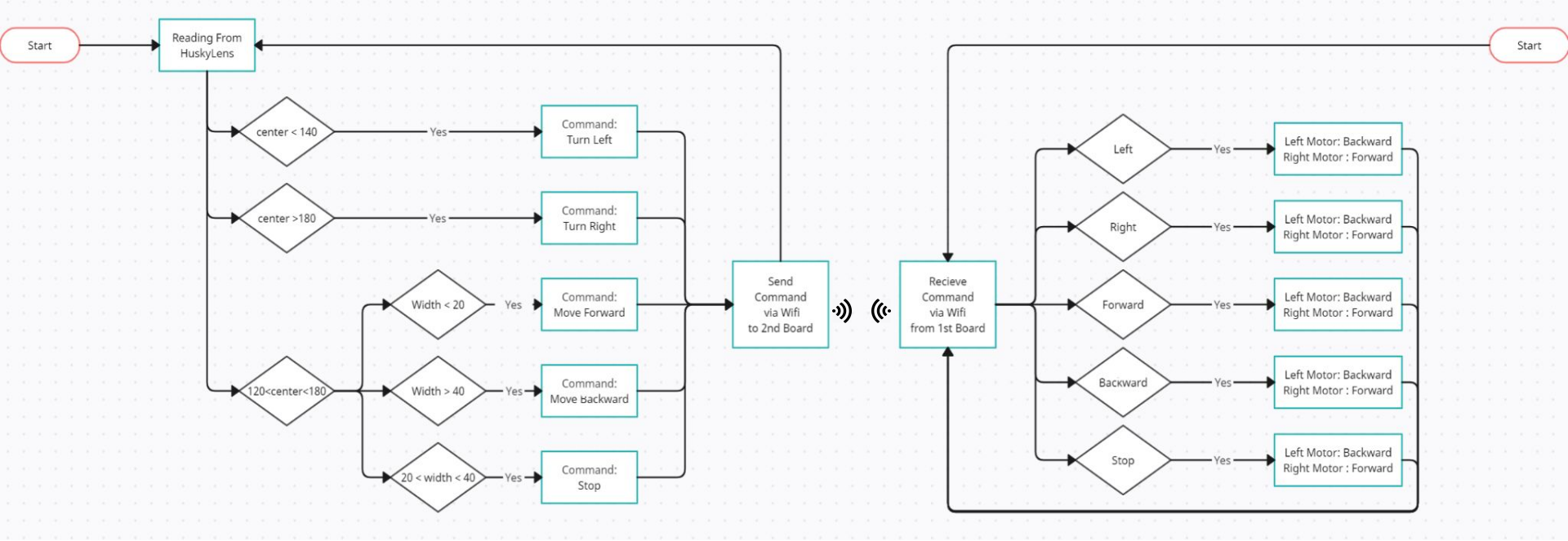
The module has a 2MP sensor and a built in screen with 320x240 resolution.

In the object tracking mode, an object is locked within range and then the Huskylens detens the object whenever it is in its zone of vision.

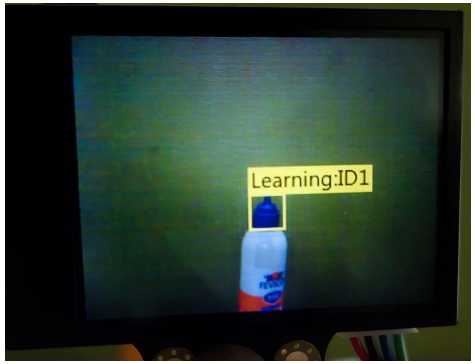




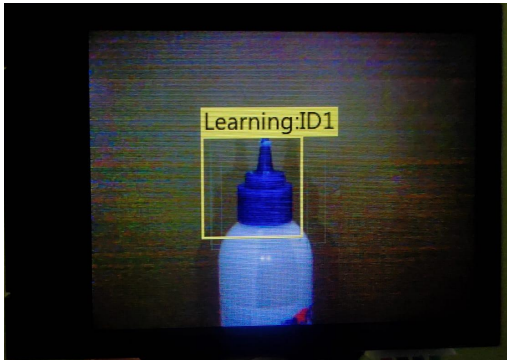
Block Diagram



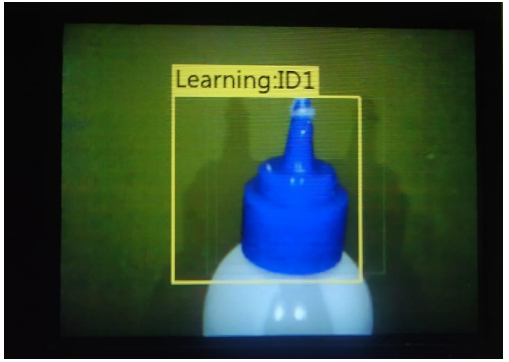
HuskyLens Images



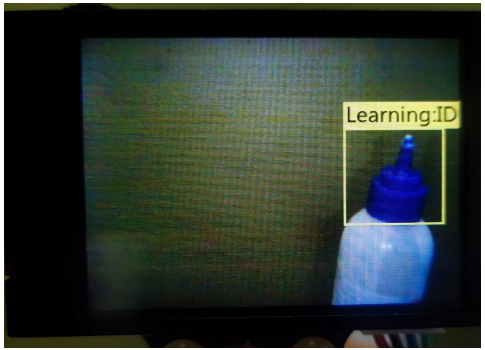
Frame Too Small (Object Far)
Move Forward



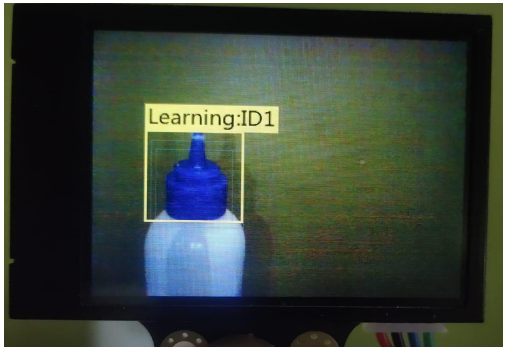
Object within desired range
STOP



Frame Too Big (Object Close)
Move Backward



Object towards right
Turn right



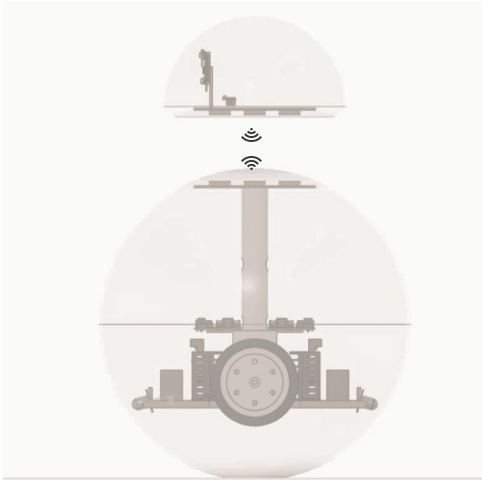
Object Towards left
Turn Left

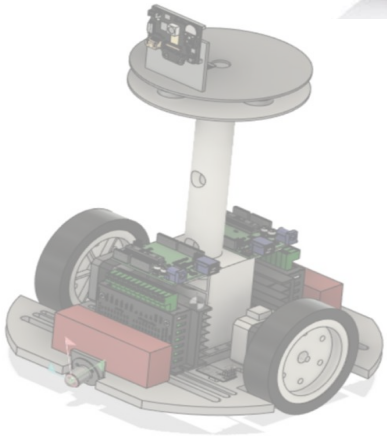
As of now ...

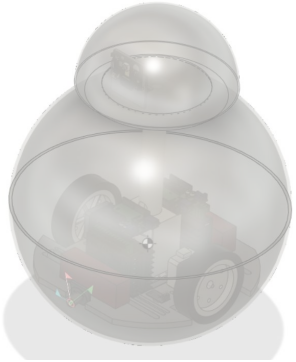
1) Ideation (Completed)
2) Cad Modelling
3) Algorithm (Complete)
4) Item Procurement

Up Next

1) Simulation
2) Prototyping & Testing
3) Code optimization

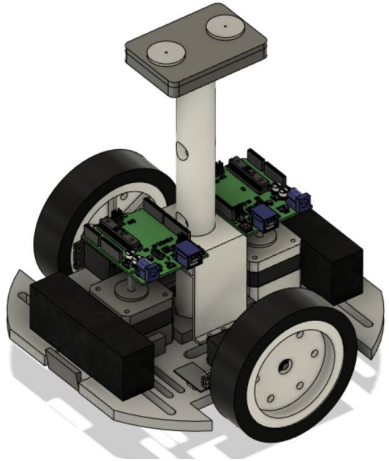






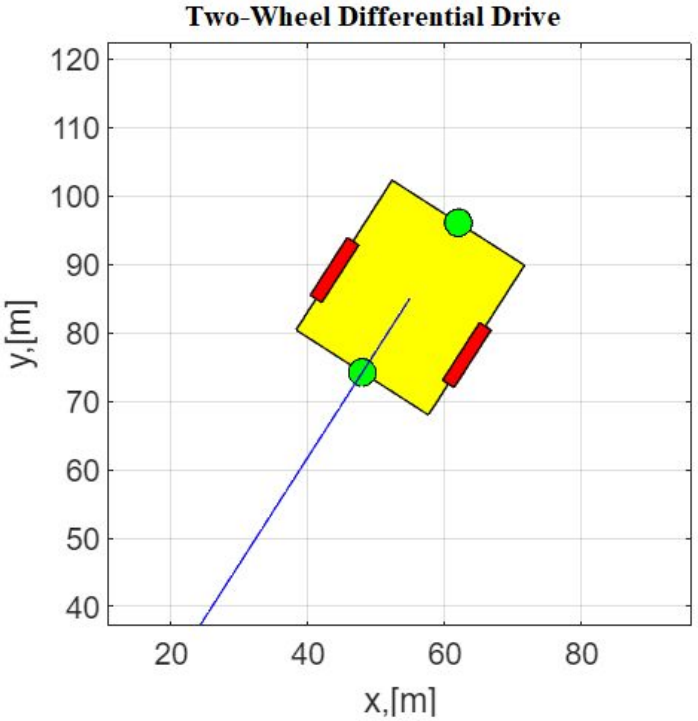
END-SEM PRESENTATION

CAD MODEL



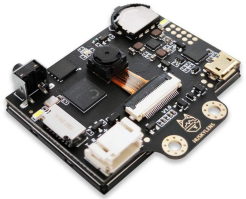
WHEEL CONFIGURATION

- The internal vehicle of the BB-8 robot has a two-wheeled differential drive configuration.
- With two castor wheels at the mid-ends of the vehicle base
- The entire robot is non-holonomic



Components

- HC05 Bluetooth Module
- MG996R Servo Motor
- Servo Motor Driver
- Neodymium (25mmx5mm)
- Ball Castor wheel
- Wheel 100mm
- LIPO Battery (11.1V, 2200mAh)
- Arduino UNO
- HuskyLens AI Camera
- Zip Ties



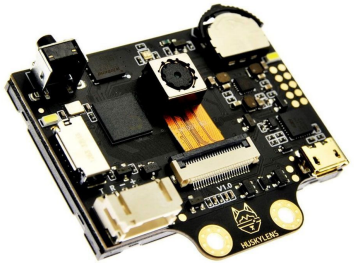
HuskyLens

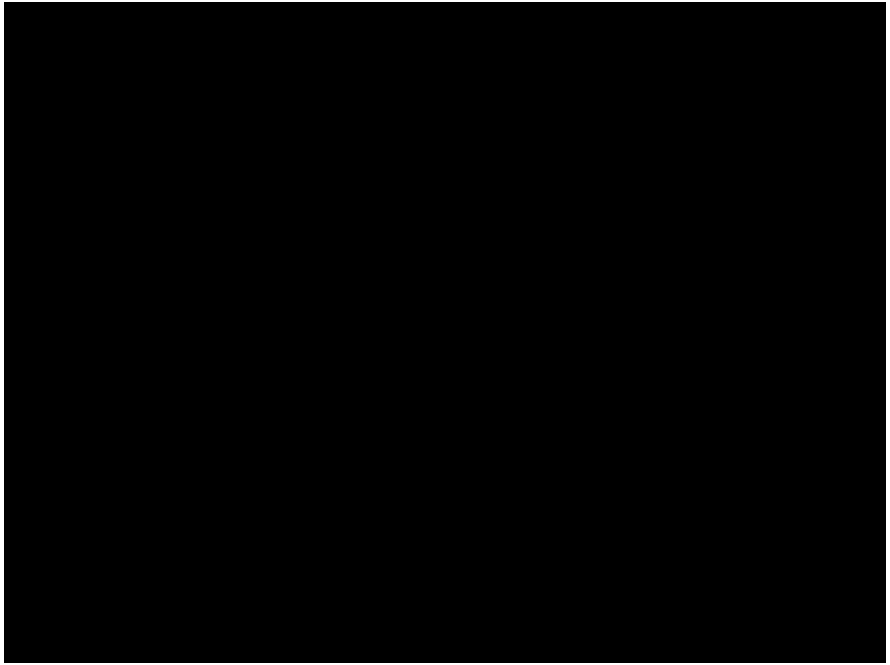
HuskyLens is an easy-to-use AI machine vision sensor. It can learn to detect objects, faces, lines, colors and tags.

Through the UART / I2C port, HuskyLens can connect to Arduino, Raspberry and other microcontrollers.

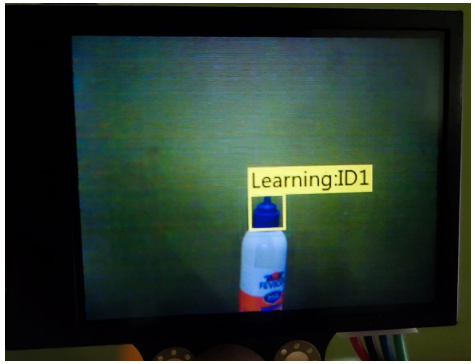
The module has a 2MP sensor and a built in screen with 320x240 resolution.

In the object tracking mode, an object is locked within range and then the Huskylens detens the object whenever it is in its zone of vision.

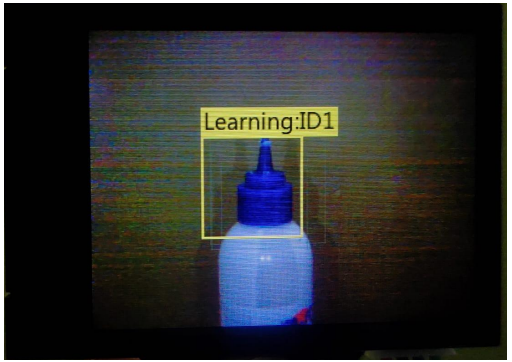




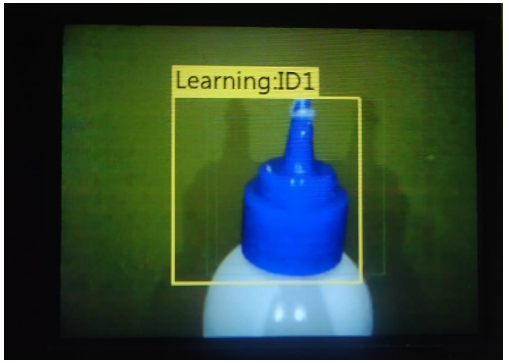
HuskyLens Images and Commands



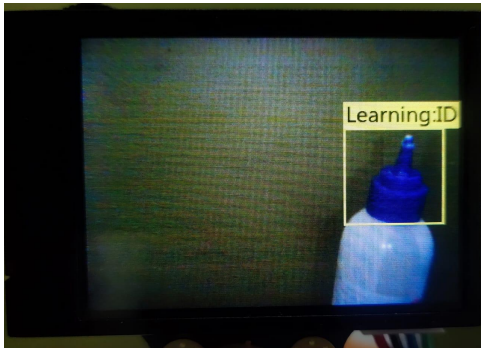
Frame Too Small (Object Far)
Move Forward



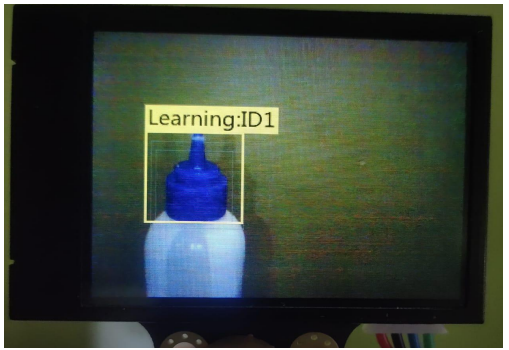
Object within desired range
STOP



Frame Too Big (Object Close)
Move Backward

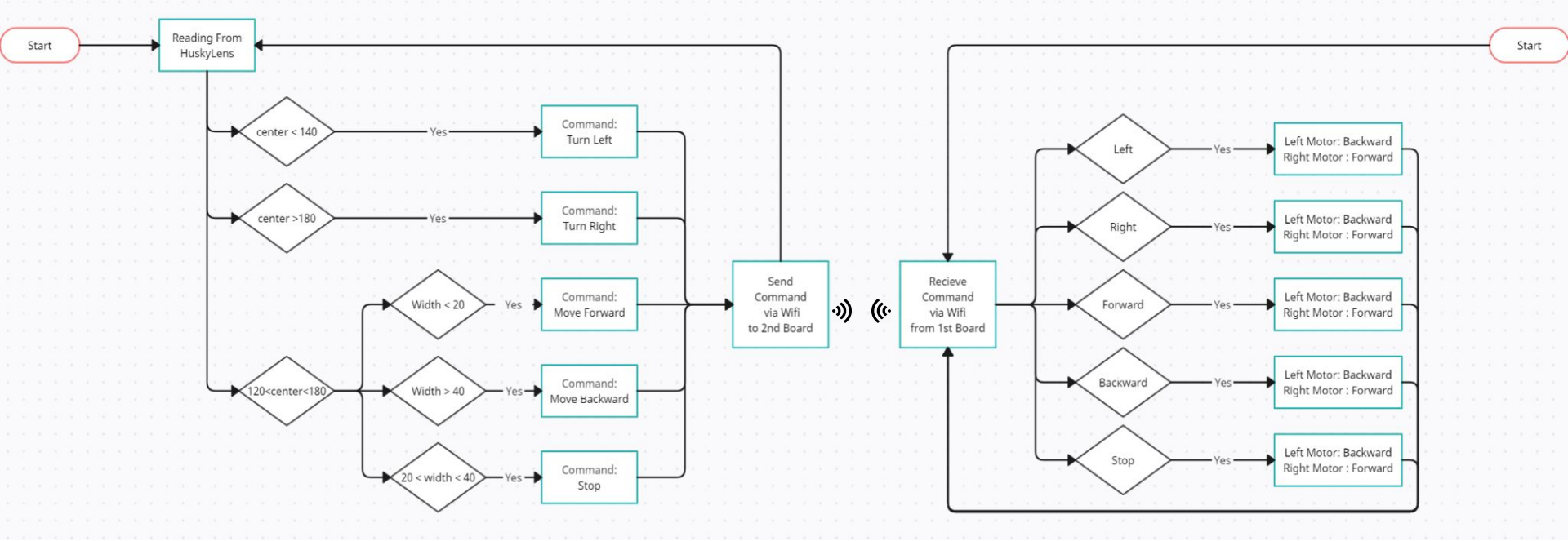
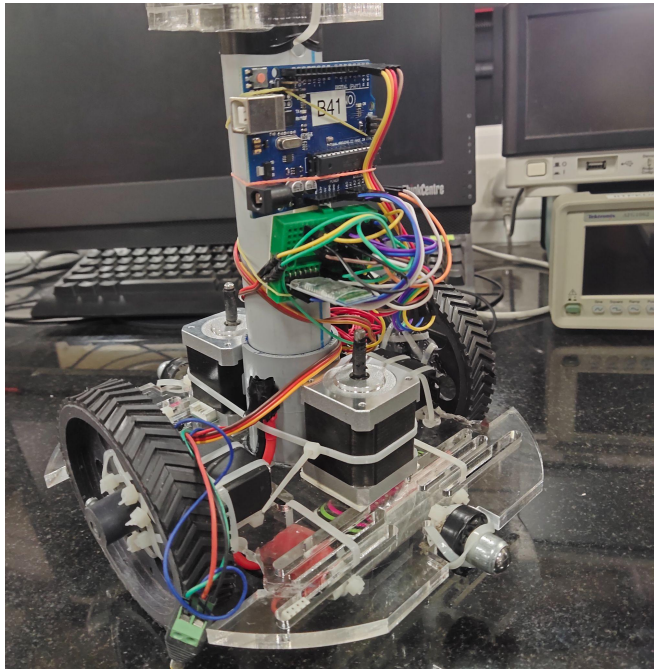


Object towards right
Turn right



Object Towards left
Turn Left

Working Code Block Diagram



Prototype