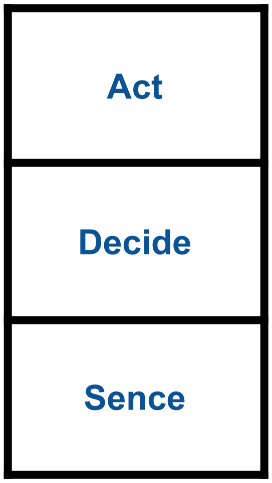
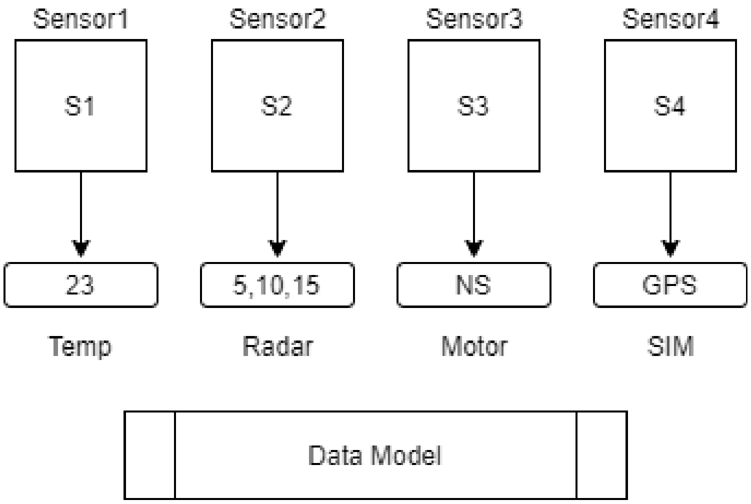
Robotics Stack



Sence

Collecting , pre-processing and fusing sensor data from multiple sources

Sensor Fusion - Ability to bring data together from different sensors and sources and combine them for the processing process.



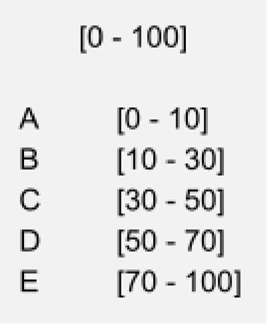
Data Model- Integrated multiple sensor data from a different source

Decide

Process the collected and created data model and apply algorithms

Data:

- Categorical Data [Classification]
  Motor Speed from 0 to 100 can be grouped into 5 groups



[0 - 100]

A	[0 - 10]
B	[10 - 30]
C	[30 - 50]
D	[50 - 70]
E	[70 - 100]

- Quantitative Data [Regression]

Process information from the environment and Plan the actions to be performed in the environment.

Act

Applying the output

- Multi-dimensional
- Specific

Carry out the actions that plan on the data model.

Event

Event Listeners

- ➔ Audio [Mic]
- ➔ Visual [Camera]
- ➔ Sensory[Temp, IMU, GPS]
- ➔ Software [API Calls, Scheduled Program, MQTT]

Event Types

Audio

[Human-Voice]
[Abnormal Sound]

Visula

[Human-Detection]
[Object-Detection]
` [Luminosity Variation]
[Abnormal Vision]

Sensory

[Temp Difference]
[Abnormal Spatial Variation]
[Abnormal Orientation Variation]

Software

[API Calls]
[MQTT-Remote]

Response Mechanism

Sr. No	Response Element	Response Drive	Output Type
1	Dialogue	RASA	[Audio] Speaker
2	Graphical Expressions	Animation Player	[Visual] On-Screen
3	Eyelids	ROS EL	[Mechanical] Servo
4	Facial Orientation	ROS-FO	[Mechanical] Servo
5	Body Posture	ROS-BP	[Mechanical]
6	Body Position Change	ROS-BPC	[Mechanical]
7	Software Action	API Calls Scheduler Engine	

Response Set

Response Set ID: 23

Dialogue: NA

Graphical Expressions: //path:suspicious_2.gif

Eyelids:
ES1 : S3
ES2 : S5

Facial Orientation
P: 30
T: NC (no-change)
R: 10

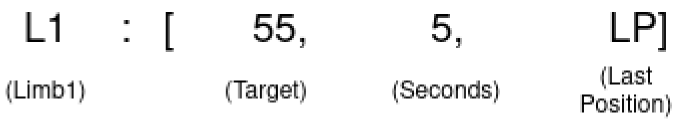
Body Posture
L1: [55, 5, LP]
L2: [55, 5, LP]
L3: [80, 5, LP]
L4: [80, 5, LP]

Body Position Change
Azimuth Angle
Destination Coordinate

Software Action: //path:sendalert.py

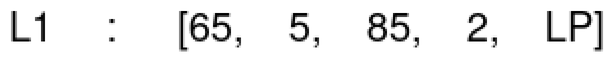
Body Posture Moments: (Data Structure)

3 Value Array(3D)



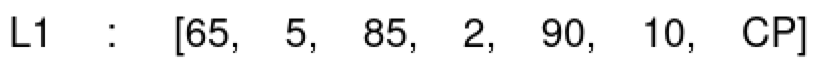
L1	: [	55,	5,	LP]
(Limb1)		(Target)	(Seconds)	(Last Position)

5 Values Array(5D)



L1 : [65, 5, 85, 2, LP]

7 Values Array(7D)



L1 : [65, 5, 85, 2, 90, 10, CP]

Deep Learning Models

- *DeepSpeech*
    - Speech-to-Text engine
    - INDIC Dataset
    - Important Links
        - DeepSpeech’s documentation
        - mozilla / DeepSpeech GitHub

- *Tacotron*
    - End-to-End speech synthesis
    - LJ Dataset

- *YOLO* (You Only Look Once)
    - Object Detection
    - Human Recognition

Trigger

Events

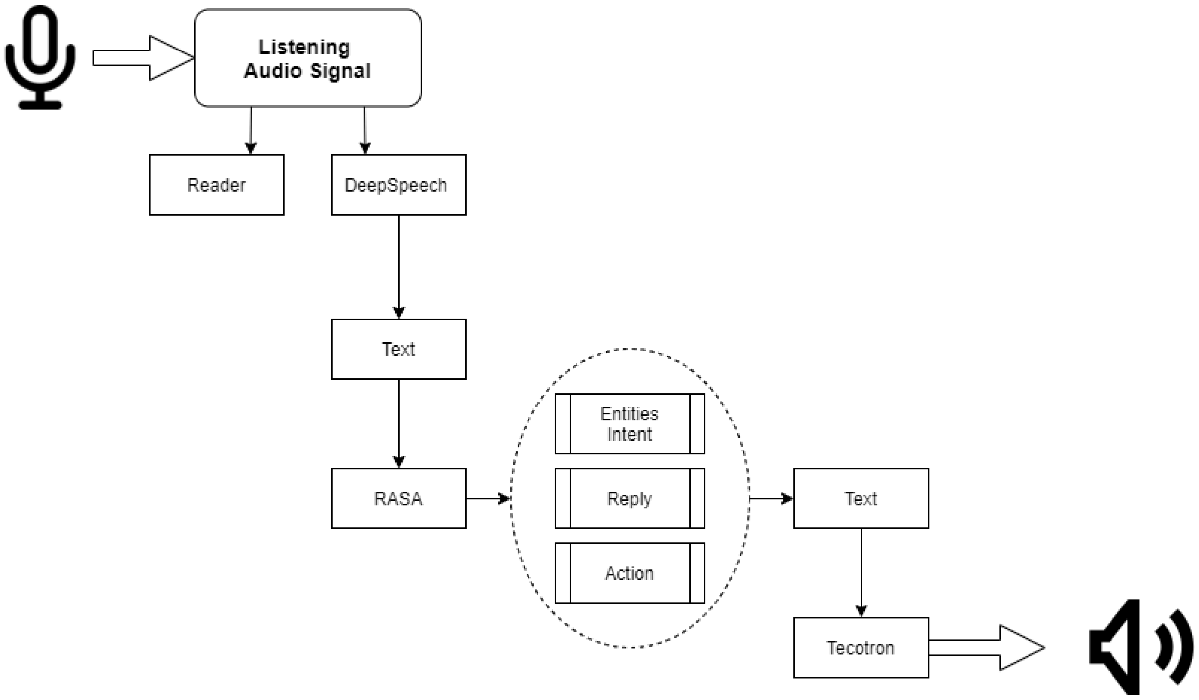
Some action take place with the help hardware and software

- Scheduled - at a particular time [time]

- Instructions - information processing [voice, visual, RC, API]

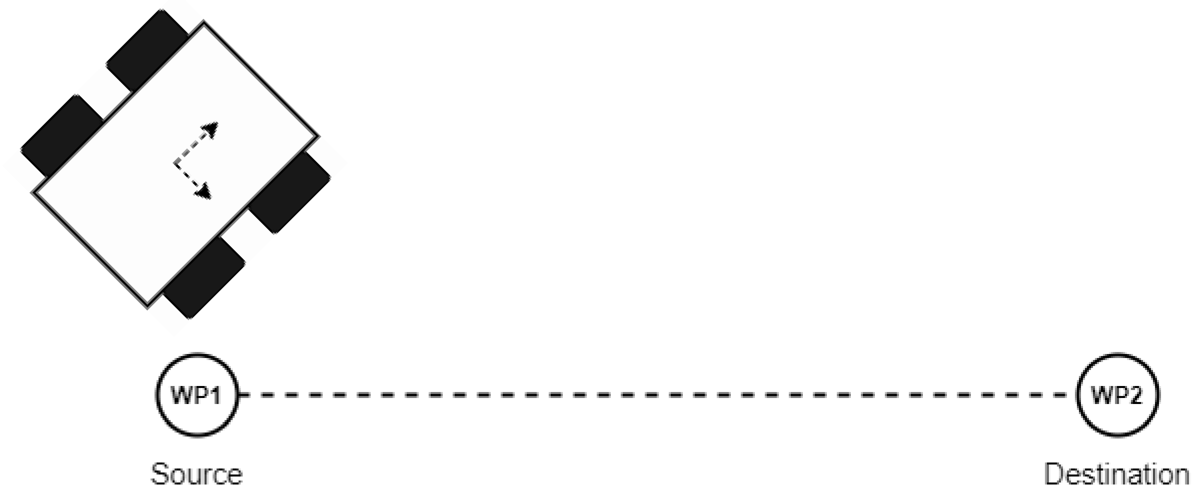
- Self-derivation - condition [instruction + logical + monitoring]

VERBAL (Audio Processing Stack)



- ➔ Sound Recording
- ➔ Hotword Detection
- ➔ Speech-To-Text [DeepSpeech]
- ➔ Speaker Recognition
- ➔ Natural Language Understanding [RASA]
- ➔ Dialogue Management (Dialogue flow) [RASA]
- ➔ Text-To-Speech (Speech Synthesis) [Tacotron]
- ➔ Sound Anomaly Detection
- ➔ Seq2SQL (Natural Language to SQL query)
- ➔ Direction of Arrival detection

Navigation



Required Hardwares:

- GPS Module
  - To get the GPS Coordinate and localise the robot in the environment
  - To get the Azimuth angle
- Digital Compass
  - To get the Delta between Heading and Azimuth angle
- Ultrasonic Sensor
  - To detect the obstacle in short range profile

Working Motors:

The robot moment in the environment will be controlled by sending the velocity command on the motors.

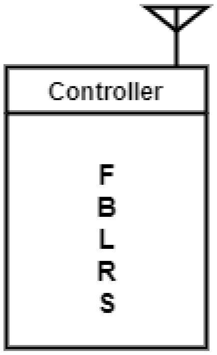
- MX1
- MX2
- MX3
- MX4

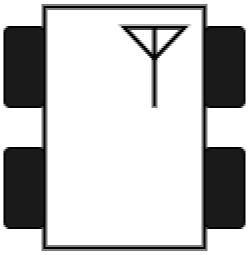
High Level Model for Autonomous System:

This model helps to control robot motion in dynamic or global environment

1. Derivation required (Azimuth angle & Distance)
2. Align noise to required Azimuth angle
3. Verify short-range obstacle profile
4. Verify mid-range obstacle profile
5. Initiate MX (using commands)
6. Verify mid-range obstacle profile
7. Verify Azimuth profile

Manual Control:





Autonomous Mobility Software(AMS): [Decision making motor controls]

I. Lane Awareness
II. Lane Discipline
III. Speed Control
IV. Obstacle Avoidance

AMS versions:

- In-house / on- premise
- Road

Lane Awareness:

1. Understand road scene

Lane Discipline:

1. Follow road rules

Obstacle Avoidance:

1. Camera
   a. Type of obstacle
2. RADAR
   a. Distance of obstacle
   b. Angle of obstacle
   c. Speed of obstacle

Speed Control:

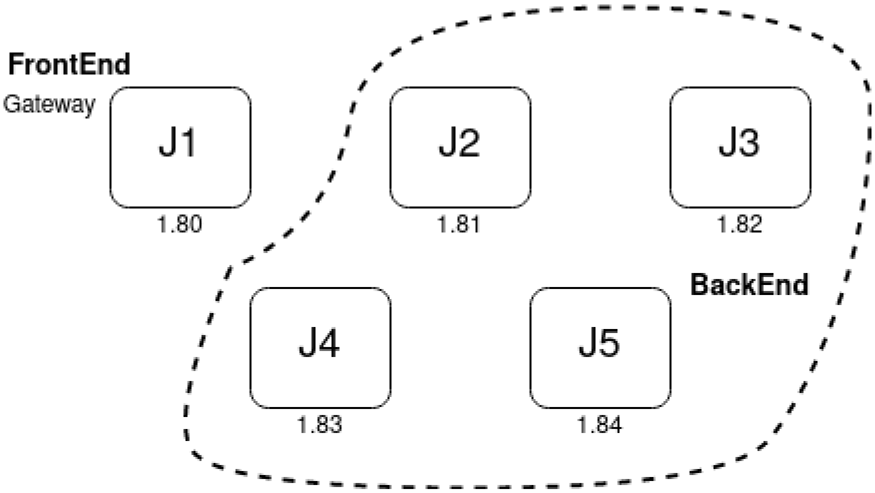
1. Selection of Acceleration Profile (AP)
   a. Distance to next position
   b. Terrain
   c. Distance to next obstacle
2. Change of AP

Task: Go to Home and Get my Pen
Process:

- ➔ Reach destination
- ➔ Action
- ➔ Return to source

1. Current location: Source
   - ○ GPS Coordinates (GPS Module)
   - ○ Azimuth angle (Digital Compass)
2. Process to reach(WP1)
   - ○ Specific to environment
3. WP1 to WP2: path detail
   - ○ Distance and Azimuth angle

Dhruyaa Compute Layout



J1:

[MQTT Broker]
[Energy Monitoring (BMS)]
[Telecom and GPS]
[Web Server / API Gateway]
[Ros Node- Publish GPS]

Connected Devices

- Bluetooth Module
  - For interfacing smarth BMS
- SIM7600H-G
  - For GPS
  - For 4G telecom
- Wi-Fi Dongle
  - For Wi-Fi telecom
- Airlink
- Satelite

Tools Installation

- Mosquitto
- ROS
- Apache
- SIM7600

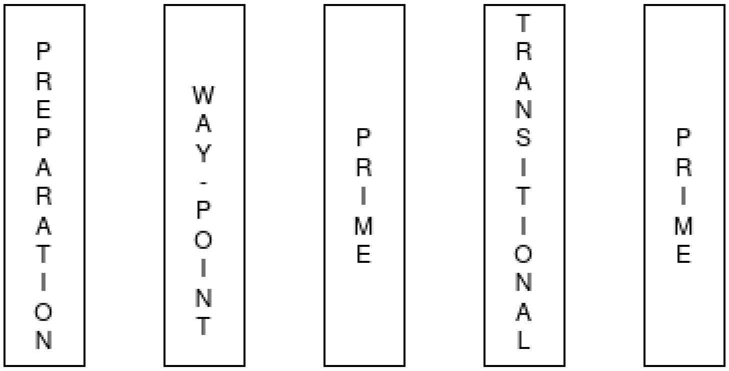
J4:

[AMS- Autonomous Mobility Software]
[ROS Node- Subscribe GPS, Publish MX]

Data Receiving

- RADAR
  - Angle/Distance
  - Speed
- CAMERA
  - Type of object
  - Depth perception
- GPS
  - Location coordinates
- Compass
  - Heading Angle
- IMU
- Ultrasonic

Decision Making Derivatives



IP Address

- Physical
- Logical

Physical

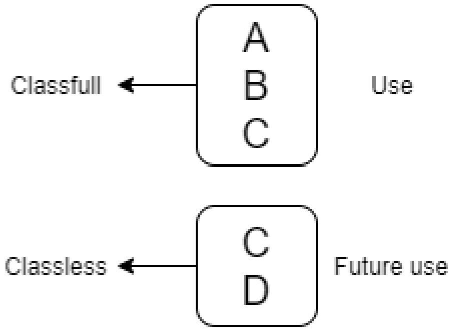
[MAC Address]
Remains Same life cycle

12 Char
6- Manufacturer
6- Device

Logical

[IP Address]
IPv4- 32 bit
IPv6- 128 bit

IP Class



A	[0-126]	N.N.N.H
B	[128-181]	N.N.H.H
C	[182-223]	N.N.N.H
D	[224-238]	
E	[240-254]	