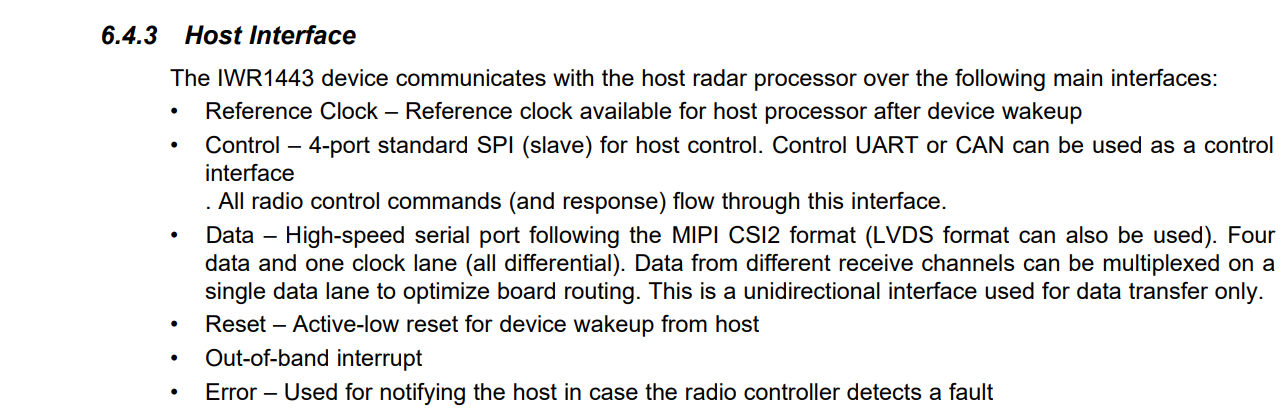
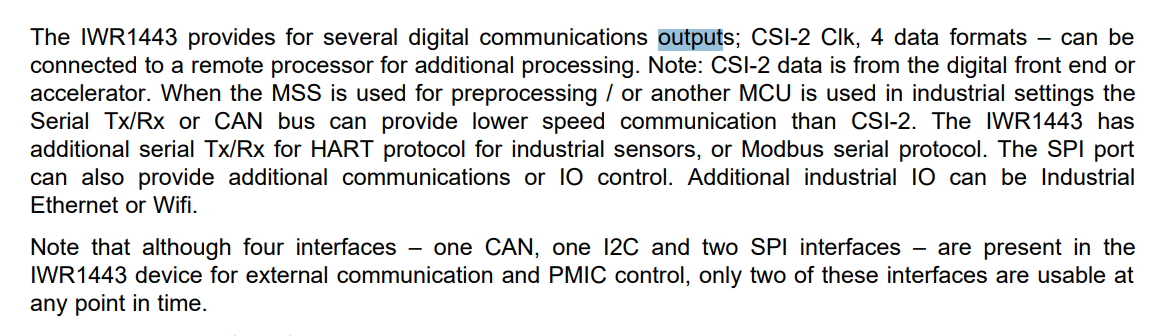
**Common approaches:**

* **Notes**
* Check that the algorithms with millimeter-wave (may behave more like visual SLAM). Figure out what data the sensor can give you and what method can make use of the data effectively.
* What do we want to show on the exhibition day?
* Study the IWR1443, the data it can output, and what SLAM algorithm is best suited for this type of data.
* System architecture (how the robot will be linked to the sensor, processing required, and if we can make use of those sensors to better our SLAM). Basically a block diagram of what the expected inputs/outputs are. Show how the robot will create a map given this data.
  + Something to discuss with Kevin (ideally by next meeting 24th May)
* ToA (Time of Arrival), TDoA (Time-Difference of Arrival), AoA (Angle of Arrive), DoA (Direction of Arrival)
* Point/Node clouds
  + Objects detected by merging individual points into a cluster
* EKFSlam and FastSLAM are common approaches to creating state space models of the landmarks and robot positions (**does it work with mmWave?**)
  + 3 main approaches for slam are PF (particle filter), KF (Kalman filter), and EKF (extended Kalman filter)
  + EKF SLAM consists of a map with large vectors stacking sensors and landmark states. It is modelled by a Gaussian variable.
  + Factored representation of SLAM and models the robot path using *particle filters*

  
The pioneer robot has sonar sensors. We could use these sensors to avoid colliding into objects (like walls etc).

mmWave outputs data in format **MIPI CS12** from the digital front end or accelerator. A widely adopted, high-speed protocol for transmission of still and video images from image sensors to application processors. Data can be read from the COM port or UART. The 2nd link has a document which shows the data which can be extracted from the mmWave sensor including objects detected (range, angle, velocity), etc (not sure exactly how to do it though, need to go in person).

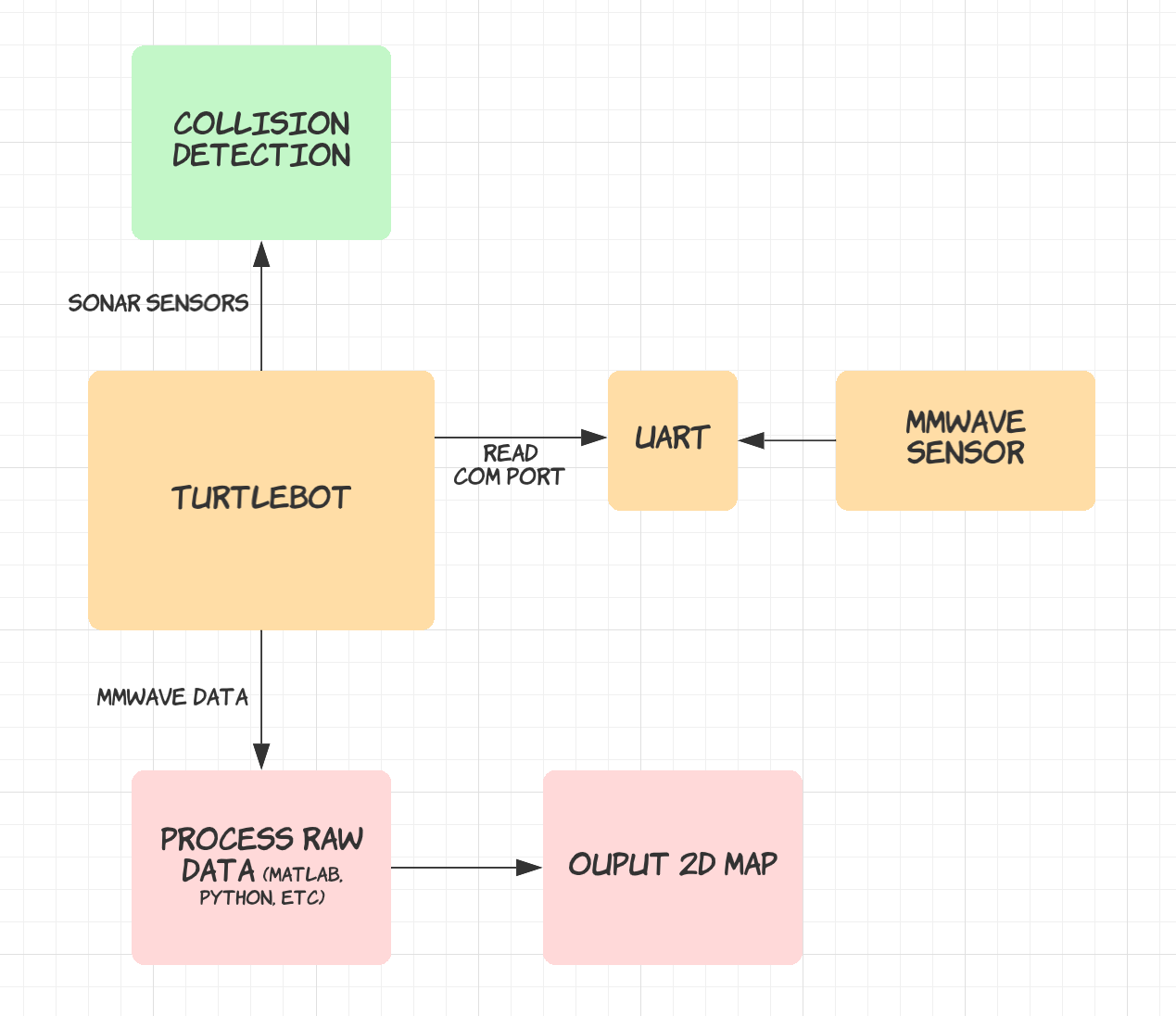
**Pioneer Robot:**

* <https://cyberbotics.com/doc/guide/pioneer-3dx>
* Nodes and API Functions - <https://cyberbotics.com/doc/reference/nodes-and-api-functions?tab-language=ros>
* <https://www.generationrobots.com/media/Pioneer3DX-P3DX-RevA.pdf>

**Iwr1443 mmWave Sensor:**

* <https://e2e.ti.com/support/sensors-group/sensors/f/sensors-forum/616564/iwr1443-interpreting-data-from-the-serial-port>
* <https://www.ti.com/lit/ds/symlink/iwr1443.pdf?ts=1653352668432&ref_url=https%253A%252F%252Fwww.google.com%252F>

**Overview:**



## **Setting up mmWave kit:**

* <https://training.ti.com/hardware-setup-iwr6843isk-and-iwr6843isk-ods>
* <https://www.ti.com/lit/ug/swru518d/swru518d.pdf?ts=1655437495593&ref_url=https%253A%252F%252Fwww.google.com%252F>
* <https://dev.ti.com/gallery/view/mmwave/mmWave_Demo_Visualizer/ver/2.1.0/>

## **Using ROS**

<https://www.youtube.com/watch?v=35U4JKThOFU>

## **Turtlebot**

* <https://www.turtlebot.com/>
* <https://emanual.robotis.com/docs/en/platform/turtlebot3/slam/#run-slam-node>

**Next deliverable**

* Mid-year progress video

**Localisation ideas**

* Panorama tracking

**Steps**

* First, figure out output from sensor
* Create test data
* Then split off into different parts
* Krithik: mmWave Sensor output
  + UART
  + Send/receive packets of data
* Shreya: Algorithm
  + Basic implementation - Find boundaries
  + Build on implementation
* Combine mmWave sensor output with algorithm
* Thorough testing

**2 parts**

* Reading data from mmWave sensor (Krithik)
  + First need to figure out what data is output from the sensor
  + Sending/receiving UART packets
* Creating the algorithm to process raw data (Shreya)
  + Using Matlab or other language of choice
  + Can use basic test data to test algorithm

**Questions:**

* What language is the best to use for different parts?
* What does the mid-year progress video entail?

**Reading data steps**

* <https://e2e.ti.com/support/sensors-group/sensors/f/sensors-forum/955017/mmwave-arw1843-problem-parsing-data>
* <https://github.com/ibaiGorordo/IWR1443-Read-Data-Python-MMWAVE-SDK-1>
* <https://www.youtube.com/watch?v=gUu8bUxUOxY>
* Read data from COM port using PuTTy
* Parse data output using Python
* Number of detected objects, range (m), Doppler velocity (m/s), peak value and 3D position (m) in **detObj** dictionary.
* Need to write this data into a .txt file which can then be fed into the algorithm

**MATLAB**

* <https://au.mathworks.com/help/nav/examples.html?category=slam&s_tid=CRUX_topnav>

**Grouping points algorithm**

* <https://stackoverflow.com/questions/29729356/how-to-efficiently-determine-if-a-set-of-points-contains-two-that-are-close>

**Sensor scenarios**

* Book put close to the sensor in front of it
* Moving sensor forward and backward from a plain wall
* Moving sensor perpendicular to wall
* Moving sensor left to right facing a wall
* Moving sensor up and down facing a wall
* Moving an object (a book) into and out of line of sight