* Project Timeline
* Intro: Why Ambarella?
* Hardware Overview From SDK
  + Chipset (Cortex?)
  + HArdware acceleration capabilities
  + NVP/GVP
  + Talk about relevance of beetle demoShows Ambarella device can easily run 30fps inferencing on 6 different camera views at the same time
    - NVP/GVP Usage
    - Strength lies in Image processing and Inferencing!
    - Shows that we can Integrate these image processing and neural network inferencing capabilities with ROS2:
      * More on that later:
  + CVFLOW
    - What is it?
    - How does it work?
  + Lychee OS
    - Why?
    - Issues with compatibility
    - Fedora
  + Autoware Review
  + Autoware Installation
    - Why docker?
* BUT: What did we focus on
  + Planning Simulations
    - Why? Because they show a good base for the cpu performance of ambarella
    - Why not perception simulations?
      * Hardware Accelerated Models generally tend to use CUDA Libraries
      * Ambarella Doesn’t have the capability to run models out of the box, we’ll touch more on that later
    - HOW does it work?
      * Predefined map and dummy info passed to ego vehicle
      * Note: DUmmy point clouds or map? Very important
* Benchmarking Methodology
  + Reaction Analyzer
  + Topic and Cpu Logger
* Benchmarking Results
  + Reaction Analyzer Diagram (Nodes)
  + Cpu Log Graphs (Ambarella during reaction analyzer)
* Ambarella Pros and Cons Summary
  + Ambarella Excels at image processing tasks, but w
* Introduction to cvflow/CNNGen
* Future integration of autoware with cvflow-adjusted models
  + CVFLow models in predefined formats (onnx, etc) can be converted into a version that can run hw-accelerated on the ambarella device
  + These can then be connected to autoware as a whole
  + But technically intensive (many models to integrate)