# 1, Chi tiết kế hoạch Sprint 1

* Tuấn:
  + Tìm hiểu chi tiết RTOS + demo nhỏ: 1-2 tuần
  + Tìm hiểu chi tiết phương án dự phòng: 1 tuần
    - Chú ý, RTOS rất có lợi trong điều khiển thời gian thực
    - Tuy nhiên RTOS không thích hợp, và không hỗ trợ điều khiển các module khác như động cơ hay module mạng (theo thầy Tiến – đã tham khảo)
* Quân:
  + Tìm hiểu chi tiết giải pháp, phương án sử dụng Java xử lý tiếng nói: 1-2 tuần
    - Có mô tả giải pháp + phương án sẽ dùng
  + Tìm phương án dự phòng nếu cần thiết
* Trung:
  + Kết hợp với Tuấn tìm phương hướng giải quyết PWM trên RTOS: 2 tuần
  + Tìm hiểu chi tiết phương án điểu khiển L298 + phương án dự phòng: 1 tuần
* Ánh + Tùng:
  + Tìm phương án cho nguồn: < 1 tuần
  + Tìm module kết nối mạng phù hợp + giải pháp + phương án triển khai chi tiết: 2 tuần
    - Demo nhỏ được kết nối nếu được
    - Tim phương án chi tiết điều khiển module mạng bằng ATmega

# 2, BÁO CÁO TỔNG KẾT SPRINT 1

* Tuấn:
  + Đã tìm hiểu được RTOS
  + Gặp 1 số vướng mắc về mặt điều khiển các tiến trình (chưa báo cáo rõ)
* Quân:
  + Đã có giải pháp về xử lý tiếng nói trên Android
    - Sử dụng API của Google
* Trung:
  + Đã có giải pháp điều khiển PWM trên L298 (chưa báo cáo rõ)
* Ánh:
  + Đã có thiết kế cho mạch nguồn
    - Chưa có phương án cho nguồn di động
* Tùng:
  + Đã có giải pháp cho điều khiển qua mạng
    - Sử dụng server wifi trên ESP8266-01

# 3, CHI TIẾT BÁO CÁO

* Động cơ: quay quanh trung điểm trục 2 bánh sau
  + Gọi lệnh start thì hoạt động liên tục
  + Gọi stop thì mới stop
* Phần mềm:
  + Ấn giữ nút rẽ trái
    - Khi ấn là gửi gói start 🡪 xe dừng 🡪 bắt đầu quay tại tâm trục sau
  + Nhả tay ra 🡪 xe dừng quay (đứng im)
  + Ấn đi 🡪 lúc này mới đi
* Điều khiển bằng giọng nói:
  + Nói trái:
    - Gửi start 🡪 xe dừng 🡪 bắt đầu quay tại tâm trục sau
  + Nói dừng:
    - Xe dừng quay (đứng im)
  + Nói đi 🡪 lúc này mới đi

# 4, GIẢI PHÁP 1 CHO PACKET TRUYỀN NHẬN

* Packet gồm 2 bytes như sau:
  + 3 bit đầu: loại lệnh (mode)
    - 000: điều khiển động cơ
    - 001: nhận nhiệt độ
  + 7 bit tiếp: dữ liệu gửi, bao gồm cả 3 bit lệnh start, stop
    - Bit 1/3:
      * 0: stop
      * 1: start
    - Bit 2/3: cho bánh trái
      * 0: lùi
      * 1: tiến
    - Bit 3/3: cho bánh phải
      * 0: lùi
      * 1: tiến
    - Tóm lại:
      * 0xx: dừng
      * 111: tiến lên
      * 100: lùi
      * 101: quay ngược chiều kim đồng hồ (quay trái)
      * 110: quay theo chiều kim đồng hồ (quay phải)
  + 6 bit tiếp: Reserved – dự trữ