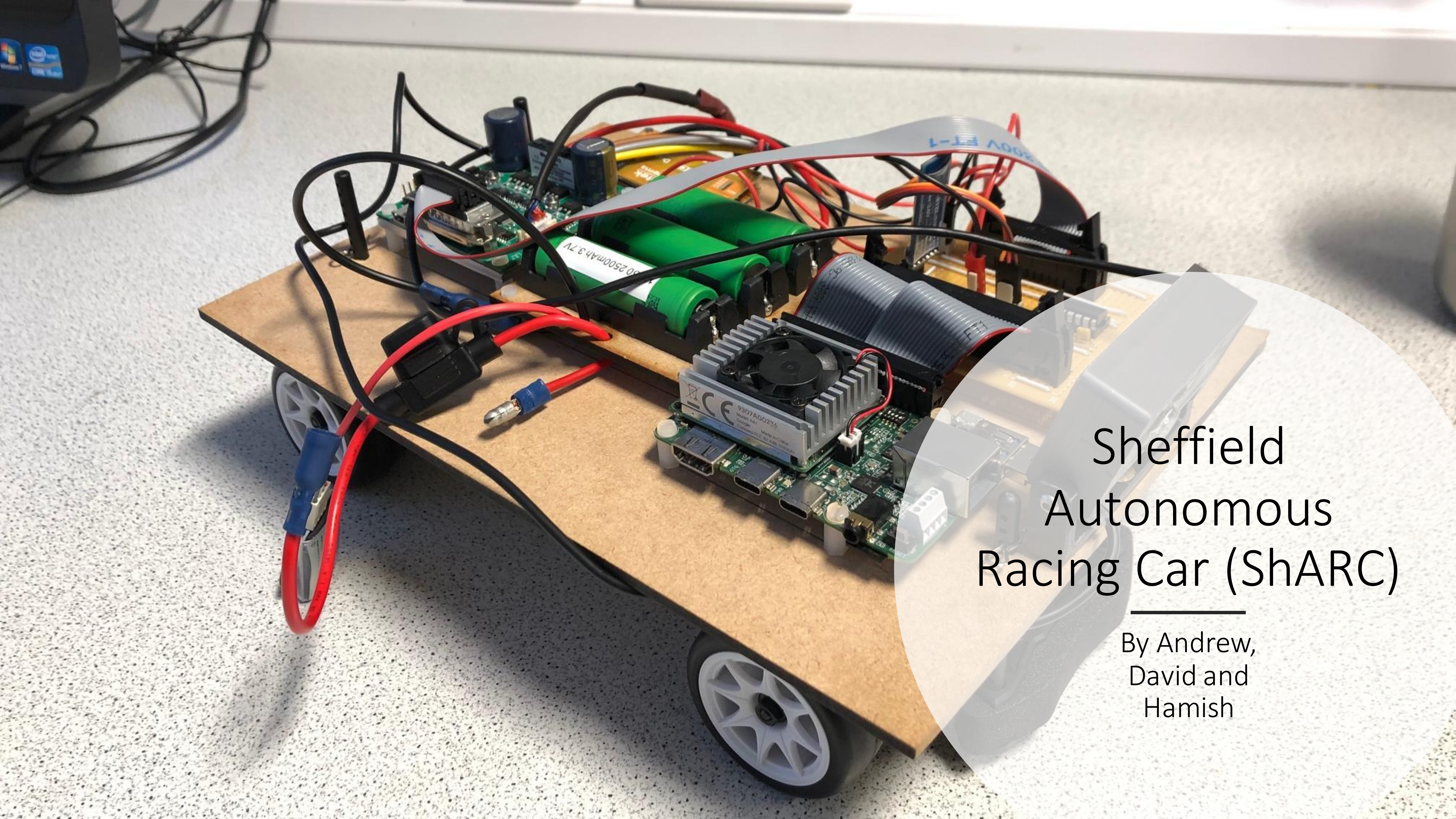


Sheffield Autonomous Racing Car (ShARC)

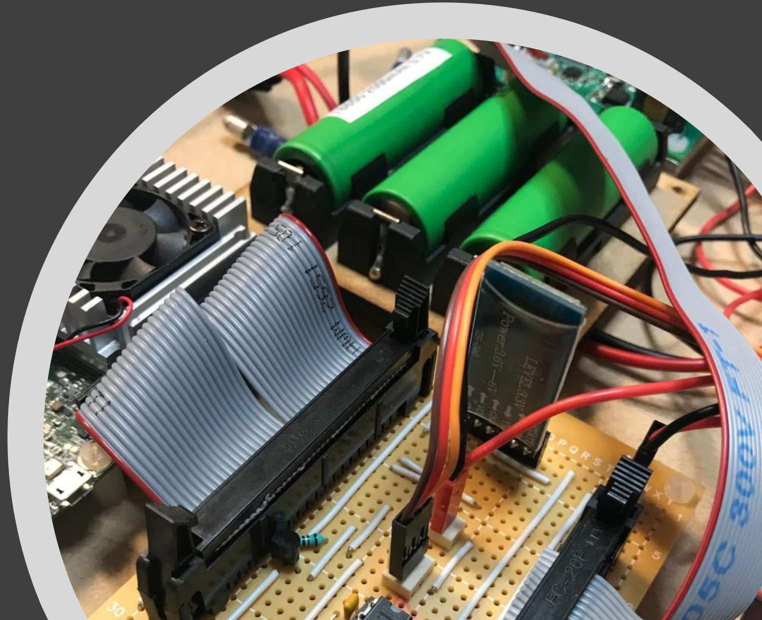
By Andrew, David and Hamish

Aims and objectives

- Implement a computer-vision based self-driving car
- Comply with key NXP rules:
  - Only NXP processors used where possible
  - Completely autonomous
  - Complies with battery restrictions
  - Primary navigation source must be a camera
  - Recognizes finish line
- Car competes in a time trial as well as challenges for additional points
  - Obstacle detection
  - Speed zones
  - Emergency stop

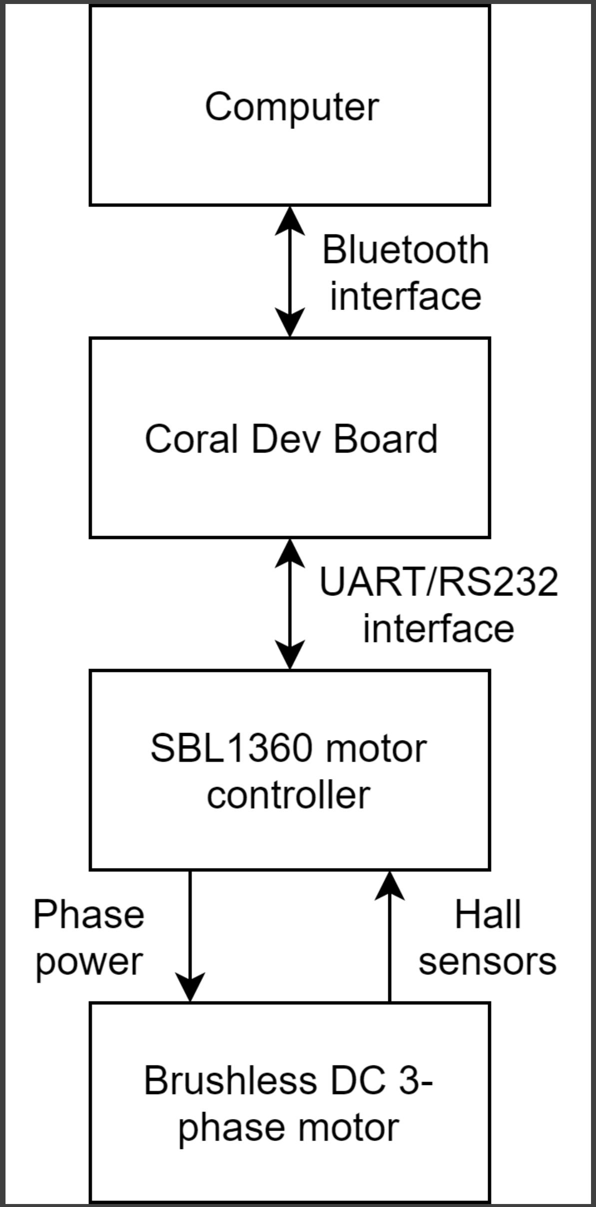
Car Hardware - Andrew

- Four-wheel drive chassis purchased and assembled
- Motor requirements:
  - Max acceleration ~9 m/s/s
  - Max speed ~6 m/s
  - 0.06 Nm at 13000 RPM
- Selected motor & gearing to meet requirements – brushless, sensored for low speed performance
- Li-Ion cells chosen to bypass competition rules on Li-Po
- Made UART to RS232 conversion, PWM level shifting, Bluetooth serial communication & power distribution board



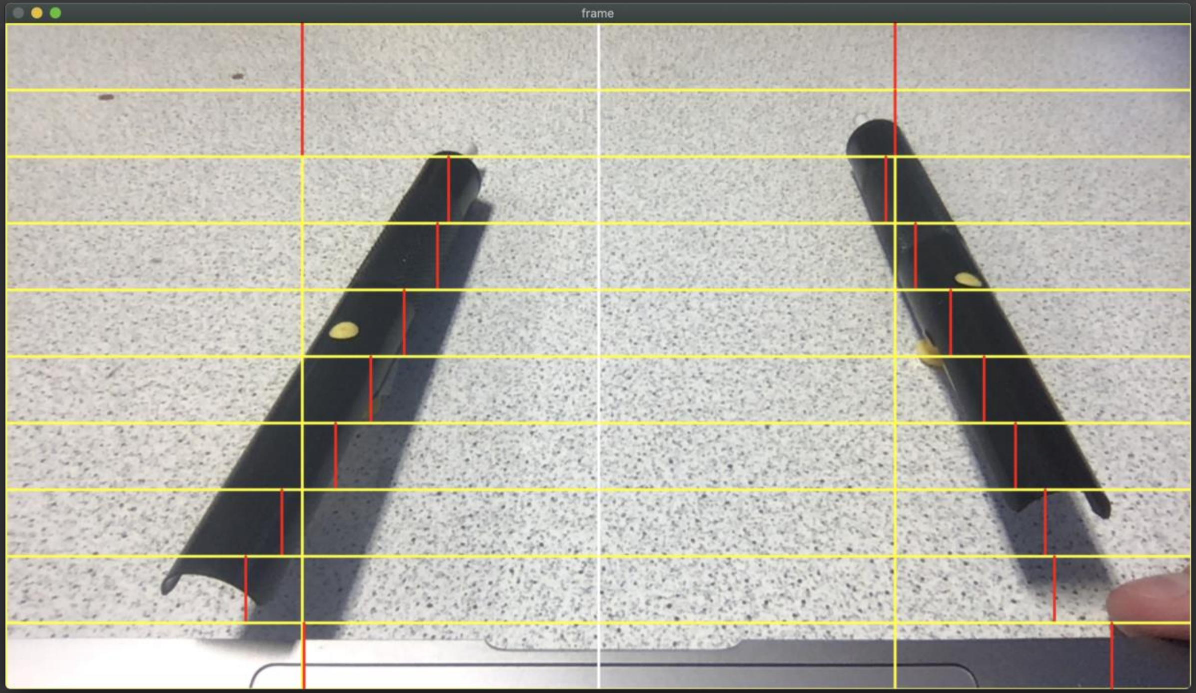
Motor control - Andrew

- Control requirements:
  - Closed loop speed
  - Dynamic braking (regenerative or rheostatic)
  - Torque/current limiting
- Would be out of scope to design from scratch, objective is to win competition
- Selected Roboteq SBL1360– regenerative braking and extensive ASCII RS232 interface
- Currently interfaced with Coral Dev Board– script allows remote speed control over Bluetooth



Hamish – Computer Vision

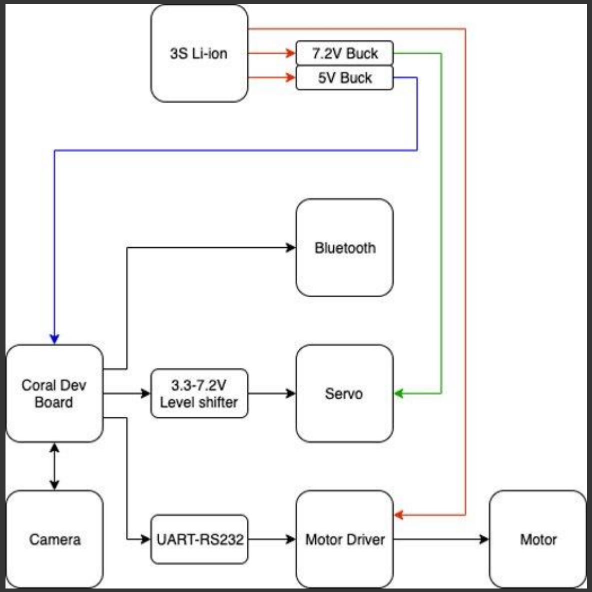
- OpenCV used
- Previously used "Hough Transform"
- Now uses:
  - Gaussian blur
  - Linear Thresholding
  - Pixel averaging



Hamish – Main Processor

- Running Linux distro (Mendel)
- Controls:
  - Servo – PWM
  - Motor Driver – UART
  - Wireless Terminal – Bluetooth

- Little online support

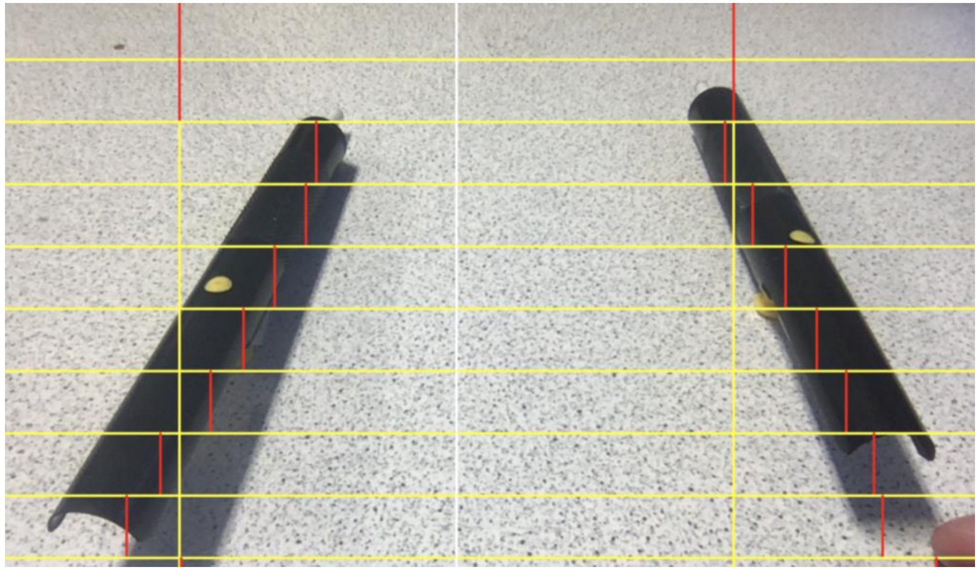


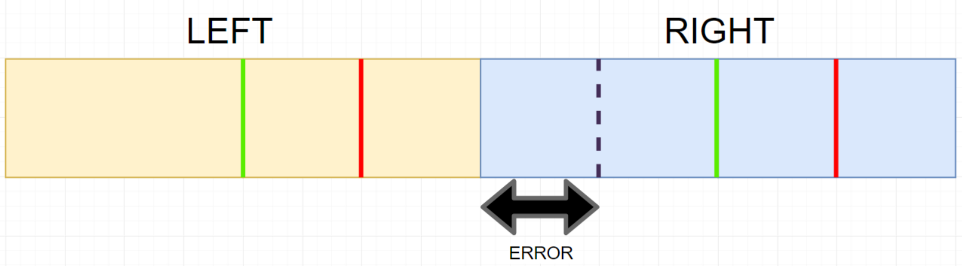
David - Steering Control (Hardware)

- Trackstar TS-920:
  - Designed for 1/10 scale trucks and 4WD buggies
  - High Torque at 13.1kg.cm (@7.4V)
  - Metal geared
  - Digital over Analogue

David - Steering Control (Software)

- Averaged vs fixed look ahead method:
  - Control via PWM
  - Averaged – receives a single value of the average of all the bars
  - FLA – only receives the data for one bar a fixed distance ahead
- Proportional Controller
  - Pixel distance used for error calculation
  - Error range mapped to servo response
  - Green/Yellow vertical lines represent track lines when centred
  - Desired midpoint shown at the centre





Progress compared to plan

- Currently one week in front of schedule – testing started this week
- All preliminary tasks completed
- Car fully operational with prototype testing platform
- Car able to follow a simple track

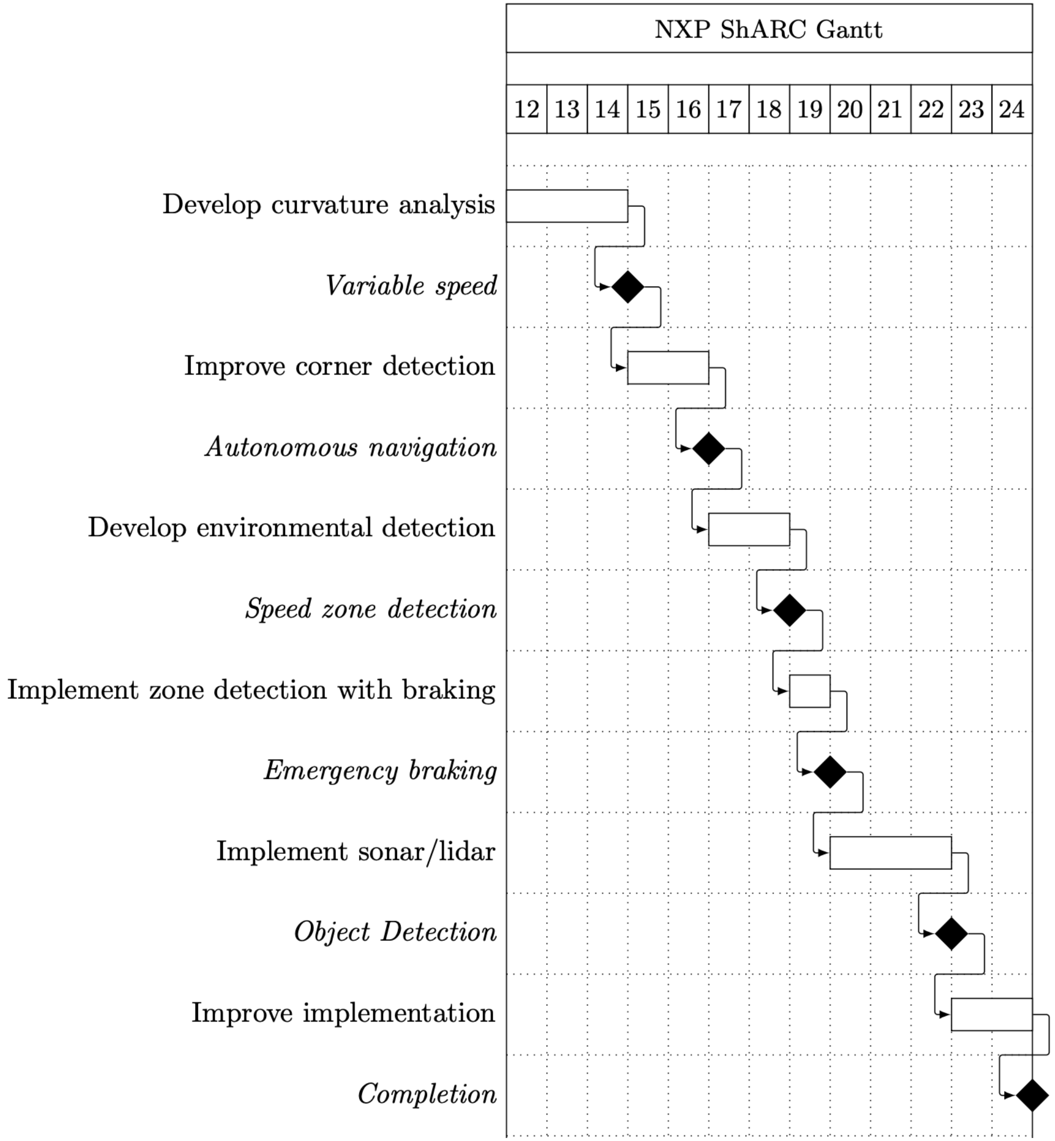
Task	4	5	6	7	8	9	10	11	12	13	14
[ALL] Complete PID	*										
[ALL] Research/order processors, camera, chassis, and wheels											
[ALL] Configure compute module and camera, assemble chassis											
[AH, DC] order servos, motor & controller based on chassis & wheel grip investigation											
[HS] Create initial line detection algorithm											
[DC] Implementing initial steering control											
[AH] Implementing initial speed control											
[ALL] Interfacing processor, speed controller, servos & adding to car											
[ALL] Interim presentation								*			
[ALL] Testing car and refinement											

Future Work

- Motor Control - Andrew
  - Speed algorithm based on line curvature
  - Tune PID
- Computer Vision - Hamish
  - Improve line averaging
  - Develop neural network approach
  - Reduce instruction loop delay
- Steering Control - David
  - Reduce over and under correction

Milestones

- Different NXP challenges:
  - Variable speed driving
  - Autonomous track navigation
  - Speed zone detection
  - Emergancy Braking
  - Object detection



Demo

https://1drv.ms/v/s!Alk-KuIpr33kgbtxgrCQoJNWZxYPgA