

AuE 835

Automotive Electronics Integration

PROJECT 1: EMBEDDED SYSTEM AND AUTONOMOUS VEHICLES

Project Schedule

- ❖ Oct 18 - Arduino and programming
- ❖ Oct 23 - Ultrasonic sensing, vehicle control & Project 1 Announcement
- ❖ Oct 25 - Signal processing review and Project 1 hands-on
- ❖ Oct 30 - Control review and Project 1 hands-on
- ❖ Nov 1 - Project 1 debugging, Q&A, Test details
- ❖ Nov 8 - Project 1 Test

- ❖ Nov 13 - Autonomous boat control & Project 2 Announcement
- ❖ Nov 15 - Project 2 debugging, Q&A, Test details
- ❖ Nov 20 - Project 2 debugging, Q&A, Test details
- ❖ Nov 27 - Project 2 Test

- ❖ Presentations and report writing

Project 1: Autonomous Vehicles

- ❖ 1.1 Signal Processing of Ultrasonic Sensors
- ❖ 1.2 Design and Control of Autonomous Vehicles
(Run Down The Hill!)

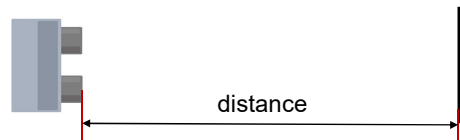


3

Project 1.1: Signal Processing

❖ Tasks

1. **Sampling:** Sample the distance data using the ultrasonic sensor(s) (frequency ≥ 10 Hz)
2. **Filtering/Fusion:** Design and implement appropriate filters to filter noises (Kalman filter is mandatory). You can either use one ultrasonic sensor or fuse multiple sensors for achieving a better precision.
3. **Calibration:** Design an appropriate fitting function to map the sampled data to real distances in millimeters and calculate the parameters in the fitting function.



4

Project 1.2: Run Down The Hill



Project 1.2: Run Down The Hill

❖ Tasks

Drive an RC vehicles to run down the ramp at CU-ICAR.
Design controllers based on ultrasonic sensors to achieve the following functions:

1. **Adaptive Cruise Control**

- Keep the vehicle 30 cm away from obstacles ahead
- If obstacles ahead are stationary, vehicle should stop at 30 cm away from the obstacles

2. **Autonomous Lane Keeping**

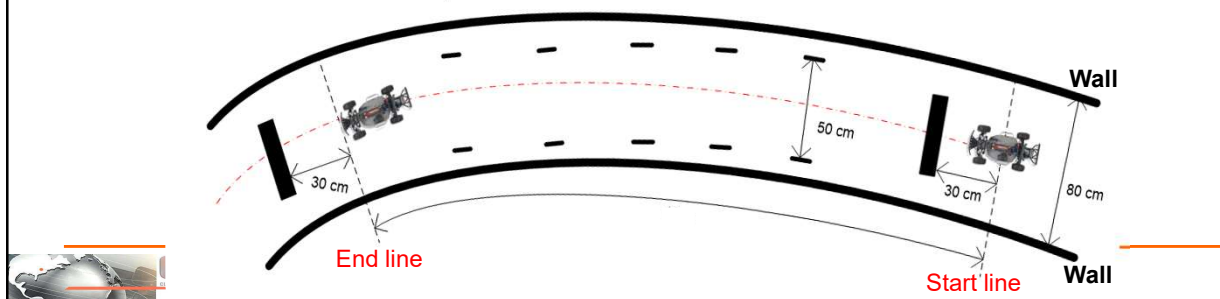
- Keep the vehicle along the center of the ramp



Project 1.2: Run Down The Hill

❖ Tasks

1. At the beginning, a board is put 30 cm away from start line and the vehicle crosses the line. Then, the vehicle should move backward to stop at the start line.
2. After the board is removed, the vehicle should start to run down the ramp along the center of the ramp (pass through 5 check points)
3. A board is placed 30 away from the end line and the vehicle should stop at the end line



Project 1 Test

1. Each group have to upload codes to canvas before test class
2. Project 1.1: Report three different distances (<1.5 m) in millimeters when TAs change the board locations before running down the hill
3. Project 1.2: Run down the hill to perform the adaptive cruise control and lane keeping tasks

Project Schedule

- ❖ Oct 18 - Arduino and programming
- ❖ Oct 23 - Ultrasonic sensing, vehicle control & Project 1 Announcement
- ❖ Oct 25 - Signal processing review and Project 1 hands-on
- ❖ Oct 30 - Control review and Project 1 hands-on
- ❖ Nov 1 - Project 1 debugging, Q&A, Test details
- ❖ Nov 8 - Project 1 Test

- ❖ Nov 13 - Autonomous boat control & Project 2 Announcement
- ❖ Nov 15 - Project 2 debugging, Q&A, Test details
- ❖ Nov 20 - Project 2 debugging, Q&A, Test details
- ❖ Nov 27 - Project 2 Test

- ❖ Presentations and report writing



9

TA/Grader Hours

- ❖ TA/Grader hours are as follows:
 - Monday through Thursday 4 PM to 5 PM.
 - Monday and Wednesday – Rahul Prasanna Kumar (cubicle #59) rprasan@clermson.edu
 - Tuesdays – Na Liu (cubicle #35) nliu2@clermson.edu
 - Thursdays – Di Huang (cubicle #48) dhuang@clermson.edu
 - If you would like to meet TA/graders in other time (e.g., Friday), please send email to make an appointment.

- ❖ All TA and graders will be available for the whole day on Nov 6-7 before Project 1 test day (Nov 8), and also on Nov 25-26 before project 2 test day (Nov 27)



10

TA/Grader Hours

- What TA/Graders can help?

- 1. Checking the **general logic** of the codes
- 2. Offering the replacement of hardware if it's definitely broken.

Make sure you're in right direction and have functional hardware

- What TA/Graders cannot help?

- 1. Proof reading the code for any type of compiling errors such as the typos or grammar errors in the program.
- 2. Fix bugs in algorithms

```
error: expected ';' before 'delayMicroseconds'
    out_test_3(100); error: expected ';' before 'delayMicroseconds'
    delayMicroseconds(20);
```