Team: Sema4ize
Jonathan Friesen - jtf698
Christopher Friesen - cmf2536
Dung Nguyen - dtn569
Nico Cortes - ngc359

Lab 7 Report

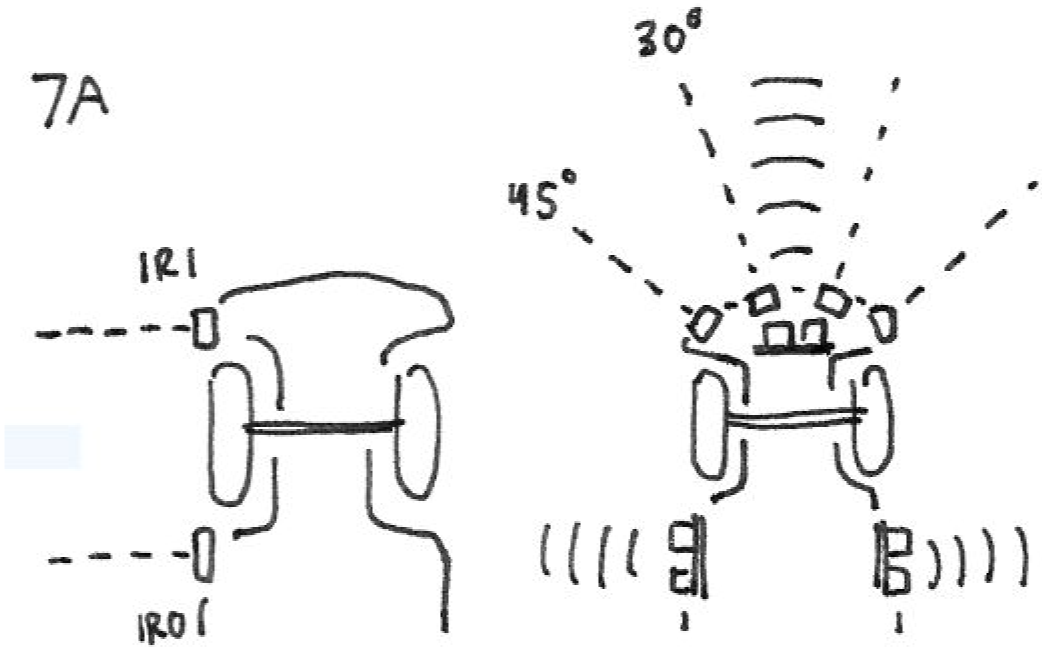
Part A

A) Objectives (not required for 7A)

B) Hardware Design

1) Rough mechanical sketch of the robot

Originally, we had 2 IR sensors on the side for Part A checkout. Our final design includes 4 IR sensors, and 3 PING))) sensors. 2 IRs are mounted at 45°, the other 2 are at 30°. All 3 PING))) sensors are mounted orthogonally.



C) Software Design (documentation and code)

1) Low-level software layer for movement and control

Controller logic is implemented in DungNico/main.c under the function “void pid()”

2) High-level test program to evaluate movement and steering

We implemented a state machine to test our turns and steering.

D) Measurement Data

1) Give the robot speed and turning accuracy

We mainly relied on trial and error to get the turn correct. We assume the turns are 90° otherwise it will hit the wall. At max speed the motors can move the robot at about 1 meter / second.

E) Analysis and Discussion (none)

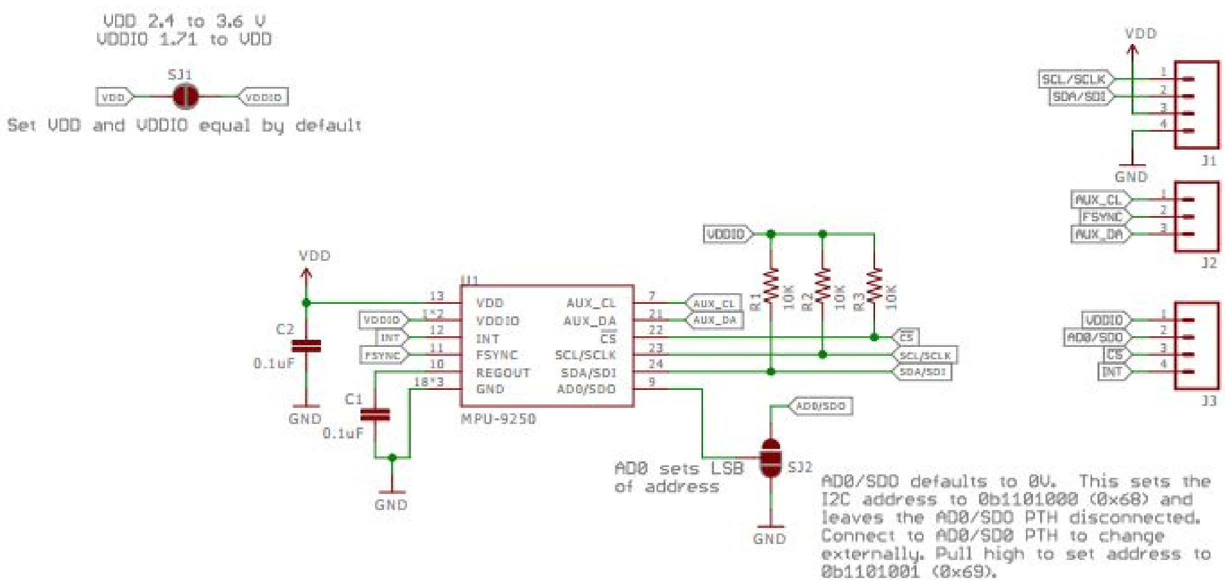
Part B

A) Objectives (not required for this 7B)

B) Hardware Design

1) Electrical circuit diagram for any additional sensor interfaces (Procedure 1)

We use an accelerometer/gyroscope/magnetometer IC breakout board. It uses the I2C interface.
https://www.sparkfun.com/products/13762



C) Software Design

1) Low-level device drivers for the sensor interfaces (header and code files)

Drivers for the ultrasonic sensor, IR sensor, and CAN are found under DungNico in US.(c|h), IR.(c|h), and CAN0*.(c|h)

The drivers were built upon Valvanoware, we had a lot of trouble with getting the PING sensor to work correctly. What we guess is going on is that sometimes the PING sensor would send a pulse to some far wall then sometime later we start new pulse. Thus, we get a false reading from the previous pulse leading to a shorter reading. We tried to mitigate the problem by extending the period between pulses as well as applying an “exponential smoothing” filter to reduce noise on the PING and IR readings.
https://en.wikipedia.org/wiki/Exponential_smoothing

2) Mid-level software layer for collision detection (header and code files)

Review Friesens Lab6 Project Lab2.c:

A periodic thread constantly reads gyroscope data registers through I2C messages, and gets the angular rate the robot is turning with respect to the “Z” Axis (parallel to the ground). If the current state is supposed to be turning but the gyroscope indicates no movement, then we confirmed a crash has occurred and back up for a period of time.

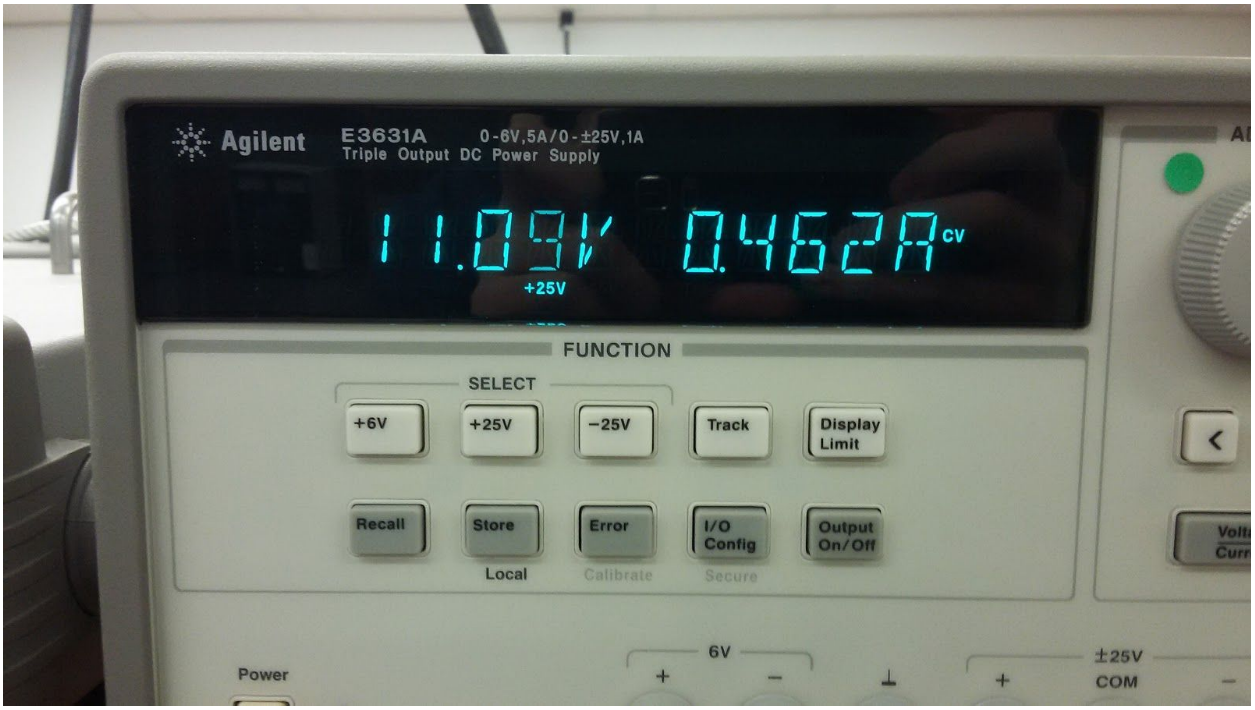
3) High-level test program to evaluate movement, steering and sensing

DungNico/main.c has some testing code that has been commented out

D) Measurement Data

1) Power supply current for various operations

Voltage and Current of one motor with no load



2) Accuracy of the positioning system, knowing where it is

We simply posted debugging information onto the ST7735 using a thread.

1) Ping data

True Dt	Measured Dt	Standard Deviation	Span
760 mm	952 mm	2 mm	2 mm
440 mm	601 mm	1 mm	1 mm
230 mm	296 mm	9 mm	3 mm
170 mm	196 mm	0 mm	0 mm
120 mm	150 mm	0 mm	0 mm

1) IR data

True Dt	Measured Dt	Standard Deviation	Span
230 mm	293 mm	3 mm	10 mm
470 mm	683 mm	1 mm	1 mm
170 mm	195 mm	1 mm	3 mm
120 mm	146 mm	1 mm	2 mm
80 mm	71 mm	1 mm	1 mm

E) Analysis and Discussion (none)

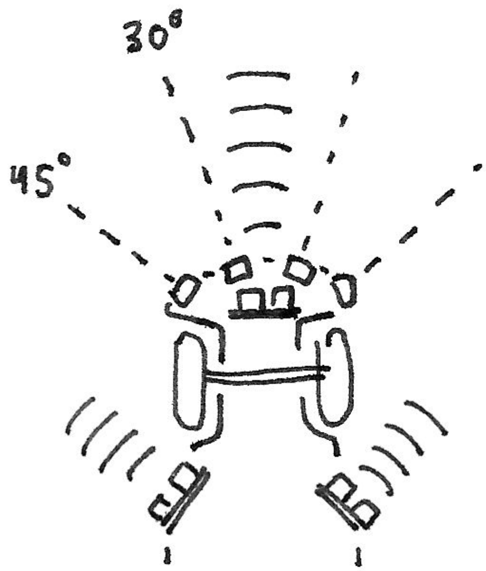
<u>Part C</u>

A) Objectives (1/2 page maximum)

In this lab we were tasked with actually implementing and testing the final control system for our robots.

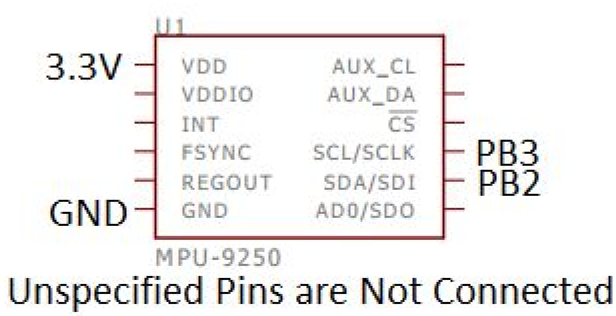
B) Hardware Design

1) Final mechanical drawing of the robot



2) Final electrical circuit diagram for the sensor interfaces

We kept everything as is except for the gyroscope ic.



C) Software Design (documentation and code)

1) Low-level device drivers for the motor interfaces (header and code files)

Refer to directory ChrisJon Project 'Lab6' /Lab2.c for all motor logic. The state machine was taken out of the motor control and put completely on the sensor board. The sensor board only sent down speeds and servo positions over CAN for the motor board to perform.

2) Low-level device drivers for the sensor interfaces (header and code files)

Refer to 7B-C

3) Low- and mid-level layers for movement, control, and detection

Review DungNico/main.c; all controller logic is in there.

Low-level stuff:

Wall angle calculation, we used basic trigonometry to calculate the wall angles. Though we have not calibrated the angles to be more correct.

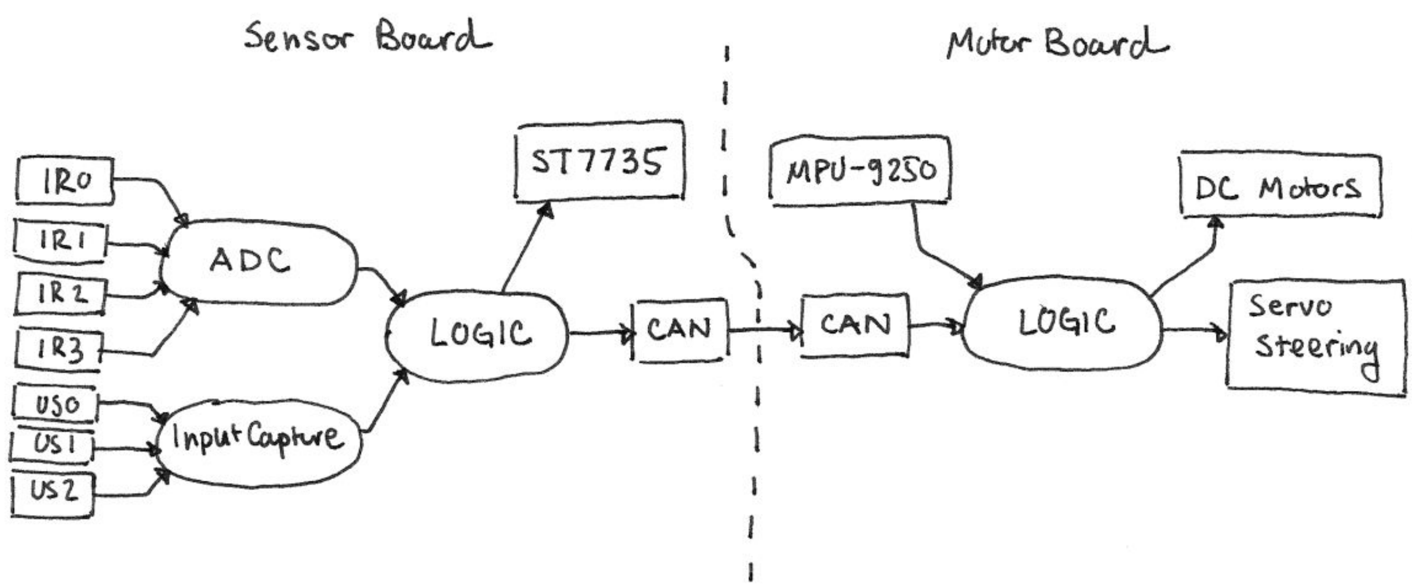
Mid-level stuff:

We used a PID to control the speed of the car. It uses the front PING sensor to detect the distance then only slows down when the car reaches a certain distance away from a wall.

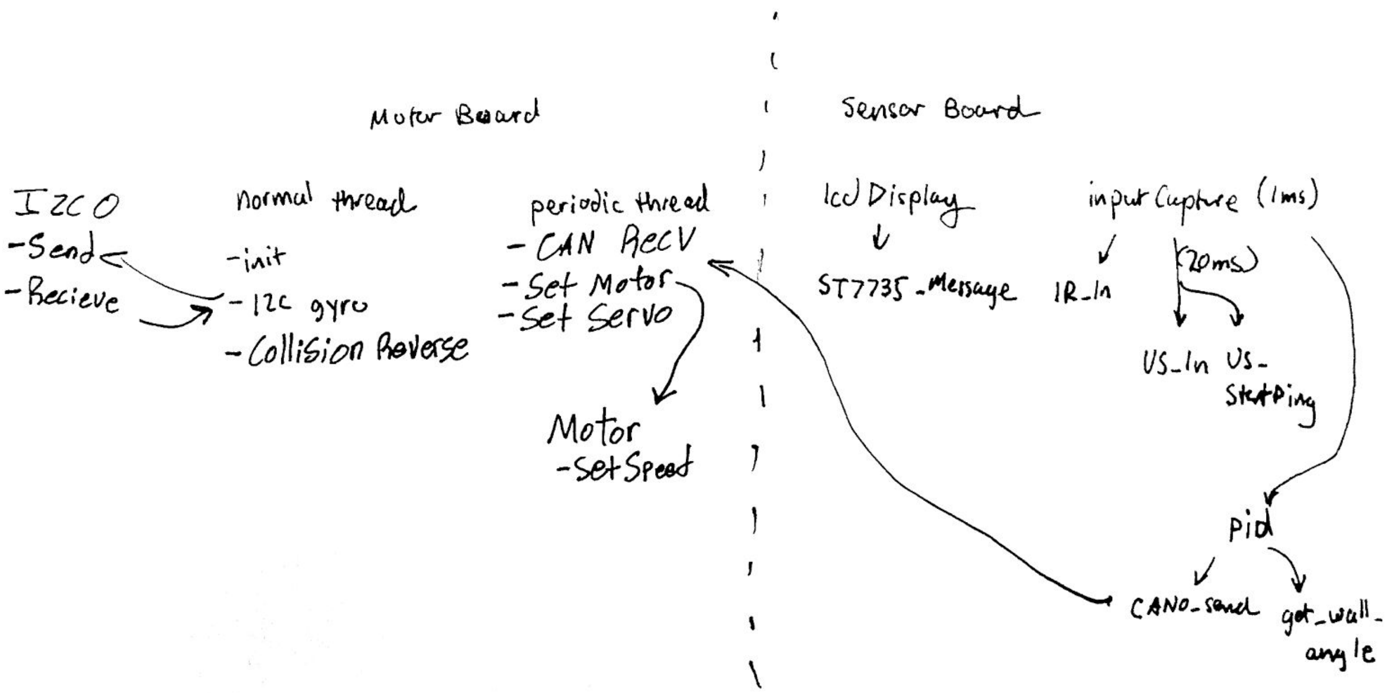
4) High-level competition algorithm

We implemented a state machine on top to preempt turns and to do various turning states. We implemented a state machine later so we didn't have much time to fully debug. The first level of the state machine decides between three conditions : too close to wall on left or right, too close to wall in front, or at a good distance. If we are too close to a the wall, the state will go to drifting away from the wall depending on which side is closer. If there is a wall in front the IR angle as well as the front US sensor determine how it will turn. If front US is clear, the robot will initiate a turn whose harshness is related to the steepness of the angle. If the front US is not clear, the robot will go into a harder turn depending on distance to the front wall. For example, if the front wall is 19 cm away, the robot will set one motor to min and the other to max in an attempt to avoid the wall. The very last state is the 'straight' state, when all sensors read that the distances are okay.

5) Final data flow graph



6) Final call graph



D) Measurement Data

1) Scores during qualifying and preliminary competitions

Two Qualifying scores were around 45

E) Analysis and Discussion (2 page maximum). In particular, answer these questions

1) What is the effect of time delay in your control system?

The control system will be less responsive. Though as long as the time delay is less than the motor constant, the performance will remain similar.

2) What sensors would you need to develop a more effective passing strategy?

Likely more IR sensors, we can get more spatial points to more accurately determine if it's a car or a wall. Hypothetically expensive LIDAR sensors would greatly improve our ability to detect cars.

4) If you hit the wall a lot, how could you have changed the design to be more effective? If your robot can travel 3 milestones without hitting a wall, you can skip this question.

We were able to move more than 3 milestones without hitting a wall, but when our robot does hit a wall it typically resulted from turning too sharply into a 90 degree turn, and we often had to calibrate that speed.

F) Post-mortem concerning team member interactions (attached to the report)

1) Each team member evaluates each other team member including oneself
Simply list one or two weaknesses.
Simply list two or three strength characteristics.

		Jon	**Chris**	**Dung**	**Nico**
Jon	Weaknesses	Not efficient code	stubborn	May not wish to stir the pot by going against ideas	Doing more work than required
	Strengths	Hard work	Won’t give up	Comes up with crazy ideas	Level-headed and great focus
Chris	Weaknesses	Stress a bit too much	Prefers to do things my way even if I know it’s bad	Sometimes offers too many alternative options for us to consider	none
	Strengths	Persistence for details	Wants to actively search for issues and solve them	Can solve seemingly difficult problems easily	Finds weaknesses in our designs
Dung	Weaknesses	Works hard	Code not spaced correctly	Sometimes gets too frustrated.	Sometimes get stuck on a single problem.
	Strengths	Works hard	Very practical solutions to problems	Devil’s advocate	Outside thinker, led us to avoid painstaking state machine only implementation
Nico	Weaknesses				
	Strengths				

2) Major failures in the way the team interacted (if any)

Accelerometer did not work as we had wanted it to, but we kept trying to use it. In the end the gyroscope worked to serve a similar crash-detection function.

3) Major successes in the way the team interacted

We work on the robot when we have time. Everyone pitches ideas on how to solve an issue. For example, before part B where the robot had to finish one lap, we were all trying to implement a state machine solution but Nico was able suggest a PID+Angle idea. We also had a solid hardware setup early in the lab which led us have more time to work on software. Our motor board never blew up!

G) Peer Review (each student submits independently and confidentially directly to the TA)

1) Classify each team member including oneself as: worked harder than average (explain), worked an average amount, worked less than average (explain)