Post Mortem for CarSim (Motion Sensing Remote Vehicle)

Project Team: Austen Chan
Brian Cheung
Bruce Wong
Wingkit Lee
Rongen Cheng

Contact Person: Rongen Cheng
ensc440-rkb@sfu.ca
rongenc@sfu.ca

Submitted to: Patrick Leung – ENSC 440
Steve Whitmore – ENSC 305
School of Engineering Science
Faculty of Applied Science
Simon Fraser University

Issued Date: April 19, 2009
Revision: 1.2

Table of Contents

Lists of Figures

1 Introduction

CarSim is a motion sensing remote controlled vehicle that adapts a natural mapping control to enhance user usability. This vehicle provides two different operation modes - user-controlled mode and automated mode. In the user-controlled mode, the user can control the car via natural steering motion from our six-axis motion sensing control. With the help of onboard sensor and processor, the vehicle will move forward and automatically steer itself around obstacles in automated mode. In addition, video streaming camera and display will be implemented on the vehicle and remote controller respectively. This allows the user to understand and visualize the surroundings of the vehicle, and hence, provides discovery functionality.

1.1 Motivation

In recent years, video driving games delivered a wrong message to kids and teenagers about driving attitudes and improper driving behaviours. Inexperienced drivers may lead to increasing rate of accidents. Risky area affects the deployment of rescue teams. All of these are the motivations of our project. Our project provides an alternative solution to all of these problems. Although our project aims to fix all of the above problems, our current solution may not be the best so far, but our future work will ensure improvements in our project.

2 System Overview

The overall system can be categorized in four different sections: signal acquisition, signal conditioning, signal processing, and mechanical components.

2.1 Signal Acquisition

2.1.1 Infra-Red (IR) Sensor

The sensing range of the Sharp IR linear sensor was from 10 to 80 cm. A 5V signal was generated when objects were sensed at 10cm, and 0.4V when objects were located outside of the sensing range, 80 cm. Signal output voltage was equally distributed between 0.4 and 5V when objects were detected within the sensing range.

There were few known issues and limitations about the selected IR sensor. Manufacturer claimed that this sensor worked better with white objects which were located parallel with the sensor. Fail to meet the requirements above would lead to inaccurate and un-stable output signal. Moreover, the output signal was not stable, and signal would fluctuate between 0 and 0.8V when no object was detected inside sensing range. Finally, a 900MHz noise pulse, which had the maximum peak at 5V, was introduced by the internal circuitry of the sensor.

2.1.2 Accelerometer

The selected accelerometer has a sensible range from 35 to 4000 times of gravity (g). This allows us to correctly calculate the impact acceleration based on the fact that one gravity (g) equals to 9.81m/s^2. The DC offset of the accelerometer is 2.5V, and it would change linearly when acceleration is greater than 35g.

2.1.3 PlayStation Dual Shock 3 Controller

A PlayStation Dual Shock 3 Controller has been selected as the user control input device with its built-in Bluetooth communication module, 3 axes (X, Y, Z) motion sensing, and analog inputs. These features fit well with our user requirements which we able to sufficiently calculate the rotation of controller, the desired throttle and brake force of the vehicle via motion sensing and analog input respectively.

2.2 Signal Handling - IR Sensor and Accelerometer

The signal processing unit, Gumstix verdex pro, did not have an analog to digital converter (ADC), and it used 3.3V as supply voltage when the rest of electronics used 5V. Therefore, input analog signals digitalization circuit and control signal (output signal) pull-up networks were needed in order to present the correct controlling and signal processing.

Accounting the maximum speed of our remote control vehicle, collision could be avoided when objects were sensed at 60cm. Therefore, a comparator circuit was added to digitalize every sensor analog output signal. As the noise was one of the known issues, a low pass filter was added to each sensor signal line for noise filtering.

The accelerometer digitalization circuit was composed with a differentiator and a Schmitt trigger. The differentiator was to monitor the occurrence of collision. A pulse would be generator at the output of differentiator to indicate that the output signal of accelerometer changed, i.e. collision occurs. A Schmitt trigger was to digitalize the pulse so that the processor would know there was a collision.

A comparator circuit was added to each control signal line to pull the control signal from 3.3V to 5V. This pull-up network could also prevent current drew back to the processor and damage it because the comparator acted as an isolator between the processor and controlled units.

2.3 Signal Processing

The signals received from the sensors, accelerometer, and the PS3 controller are feed to the Gumstix for processing. They are, then, used to generate the output signals for the driving motor, turning motor, and the camera's motor. These signals are all pulses with variable duty cycles and frequencies, and they are generated by setting and clearing the corresponding GPIO pins on the Gumstix. The enablement, duty cycle, and frequency of the output signal are determined by the input signals.

Before outputting any signals from the Gumstix, we need to give empty signals to all the output pins that we need to use (clearing all pins) to prevent burning of the circuitry. First, the controller signals are stored in a buffer array, each byte represents the status on the controller and the button signals are used to determine the mode of operation. In manual mode, we rely on the status of the movement of the PS3 controller to determine the turning angle and the L2, R2 analog buttons are used to determine the speed of the vehicle in the forward and backward direction respectively. The level of pressure put on the L2, R2 buttons are determined by an 8-bits number, where 0 means not pressed and 255 means the button is pressed all the way down. The movement of the turning is determined by using the normalized acceleration data in the x,y,z-component from the buffer and convert it to rotation in radians using trigonometry. According to the axis defined by the PS3 controller the following equation determines the turning angle of the controller assuming we use the conventional way of holding the controller:

$$\theta = -\tan^{-1}\left(\frac{a_x}{a_z}\right)$$

, where θ is the turning angle of the controller, a_x is the acceleration of the controller in the x-direction defined by the controller, and a_z is the acceleration of the controller in the z-direction defined by the controller.

The camera's motor turning is depended by the right analog stick and the R1 button. The status of the right analog stick is determined by an 8-bits number, where its value lower than 127 means the stick is moving to the left and the camera's motor will turn to the left, and when the number is greater than 127 means the stick is moving to the right and the camera's motor will turn to the right. The direction of the camera's motor is depended on the duty cycle and the frequency of the outputted signal. Lastly, to return to the center position, once the status of the button R1 gives a non-zero value the motor will return to the center position by changing the duty cycle of the pulse.

Using the similar fashion, the left analog stick controls the activation of the head lamp of the vehicle. Indeed, LED indicators are used to distinguish between modes.

In modes that requires automation, namely the collision prevention mode and the collision detection mode, the input from the sensors are required to detect obstacles around the car. The output signals for the driving and turning motor act correspondingly to the status of the sensors using the following algorithm in the following order:

- If obstacles surrounded the car, car will stops.
- If the front was blocked completely, car will move backward turning to the left; if obstacles were present on the back left, it will move backward straight; lastly; if obstacles were present on the back, it will move backward turning to the right.
- If the front was block, the car will move forward right and if the front right was blocked the car will move forward left.

During the automated mode the following algorithm is included to give the car its default motions:

- If there were no obstacles in the front and the front left, the car will randomly choose to move forward either straight or to the left.
- If there were no obstacles in the front and the front right, the car will randomly choose to move forward either straight or to the right.

Lastly, when a collision occurs during collision detection mode, the car will stop when the user is in the collision detection mode.

The detected motion of the controller not only determines the motion of the vehicle, they are also used to map out the route the vehicle has gone through using the following algorithm:

1. The motion of the vehicle at every clock cycle is recorded as a vector, namely the motion vector.
2. If the car was moving forward then the location (co-ordinates) of the car will be updated using the following formulae:

$$\theta_c = \theta_p - \theta$$
$$x_c = x_p + h \cdot \cos(\theta_c)$$
$$y_c = y_p + h \cdot \sin(\theta_c)$$

,where θ_c is the current orientation, θ_p is the previous orientation, θ is the orientation of the motion vector, x_c is the current x-value, x_p is the previous x-value, y_c is the current y-value, and y_p is the previous y-value.

3. If the car was moving backward then the location (co-ordinates) of the car will be updated using the following formulae:

$$\theta_c = \theta_p - \theta$$
$$x_c = x_p - h \cdot \cos(\theta_c)$$
$$y_c = y_p - h \cdot \sin(\theta_c)$$

The file containing the coordinates of the route is divided into different sessions, and the user can reset the coordinate by pressing L1.

2.4 Mechanical Component

The prototype Motion Sensing remote controlled car is built on the New Bright R/C Pro Dirt Buggy chassis. A new motor, and a new circuit board were used instead of those on the original car. In the most updated version of the Motion Sensing remote controlled car, two platforms are attached to the car, a sheet metal on the bottom of the chassis, and a plexiglass platform on top of the car. The sheet metal on the bottom is screwed onto the bottom of the chassis to provide support for the top platform. The top plexiglass is for the mounting of the 8 sensors, the circuit board, and the camera.

During the course of the project, the original spring for the rotational motor was out of condition to return the turning wheels to the central position. Therefore, a spring mechanism was deployed to solve the problem. The spring helps bull the wheels back to the forward position. In addition, another student while playing without remote controlled car broke the pillar support near the end of the car. The problem could not be fixed as it was badly broken and no suitable parts could be found to fix it. However, the effects of this are insignificant.

As mentioned earlier, one of the original motors was replaced, and it is the rotational motor at the front. Because the original motor could only turn at a single angle, we replaced it with a servomotor that gives us a few more turning angles, thus allowing the car to steer according to the relative angle of the PS3 remote controller. As the project goes on, more and more components are finished and are attached to the card. As a result, the weight of the car increased significantly and caused the front wheels to be out of shape, hence hindering the forward movement. To solve this problem, a pair of plastic band were used to secure the wheel chassis with the car's main chassis so that all the weight goes onto the main chassis and not onto the wheels.

3 Budget and Timeline

3.1 Budget

Table 1 consists the estimated and actual cost of our project up to April 17, 2009.

Equipment	Estimated Cost	Actual Cost
IR Sensor	$80	$111.18
Remote Car	$80	$78.39
Bluetooth Microporcessor	$100	$560.72
Accelerometer	$60	$31.32
Camera	$50	$88.48
H-Bridge Driving Motor (2 units)	$40	$26.35
Controller	$60	Free (Borrowed)
Monitor	$50	$72.8
Camera Adapter	$6	NIL (Did not use)
Processor Connector and Power Supply	NIL	$131.6
MicroSD card	NIL	$11.2
Camera Motor	NIL	$18.3
Batteries	NIL	$90.6
Miscellaneous	$78.9	$138.03
Total	**$605**	**$1358.97**

As shown in table 1, we can see our actual project cost is roughly 2 times the original estimated cost. The reason is that during the development stage, we accidentally burnt our Bluetooth processor (Gumstix) which costs $300 for replacement. Also, during testing stage, we figured we need to have more IR sensor to provide sufficient obstacle detection, as a result, a total 8 sensors have been bought instead of 2 as previously estimated. Furthermore, with the increases of sensors and their related circuitry, power consumption increased. This leads to extra cost on batteries as the original onboard battery does not provide sufficient power for all the circuitry of the system. Although the actual cost is over budgeted, roughly 10-15% of the total cost has been the shipping cost. In future, cost can be reduce at least 5% if we are able to purchase all the components in one to few stores instead of everywhere around the globe.

3.2 Timeline

Task Name	Start	Finish
Research	Thu 1/8/09	Wed 2/18/09
Proposal/Funding	Tue 1/13/09	Mon 1/19/09
Progress Report	Wed 1/28/09	Mon 2/2/09
Functional Specification	Mon 1/19/09	Mon 2/16/09
Design Specification	Mon 1/19/09	Mon 3/2/09
Software Development	Wed 1/21/09	Tue 3/24/09
Module Test and Assembly	Mon 3/2/09	Wed 4/1/09
Debugging/Prototype Modificatio	Wed 3/4/09	Wed 4/1/09
Presentation, Report and Demo	Thu 3/26/09	Wed 4/1/09

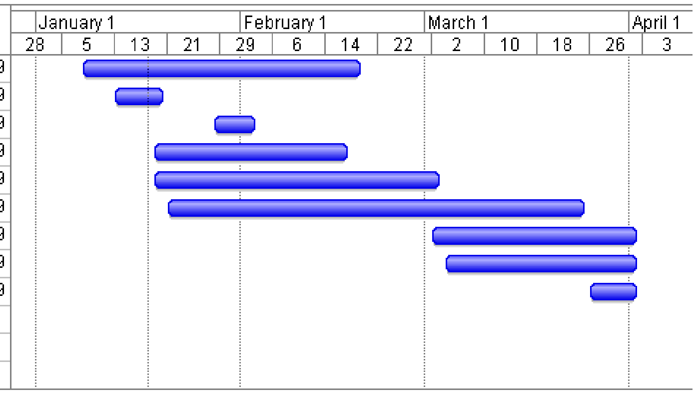


Figure 1 - Proposed Project Gantt chart

Figure 1 shows the proposed project Gantt Chart. Since this chart proposed at the beginning of the development cycle, the final implementation timeline is off by 2 to 3 weeks. For example, during the software development stage, we suggested that it should take place between mid-January to late March. Due to extra effort required on processor development environment setup, the true software development began in early March and ended early April. Luckily, on the hardware system, we do not rely on any special module to implement our own circuitry. Hence, our hardware development cycle is on track and then we were able to shift the extra work force on helping the integration and testing. Although the actual timeline differs from the proposed one, it is extremely important for us to have a Gantt Chart. The reason is that it allows us to foresee the upcoming tasks and adjust it based on the current project condition. If we do not have this Gantt Chart beforehand, this may lead to a catastrophic failure to our project.

4 Group Dynamics

Although each member were given their own parts, we also work together to resolve problems as they arise. Not only focusing on our own parts, we tried to help each other to come up with better solutions to problems that we encountered. In the early stage, we set up weekly meetings to gain group momentums. During the meetings, we came up with project ideas and allow each member to vote for their preference. We reviewed each other's work and report them during weekly meetings. Throughout the term, we constantly communicate with each other regarding new issues arose as well as updating each other with project progresses. All members were involved with project documentations and one person is responsible for proofreading to ensure coherent work throughout the documents. Near the end of the term, our group worked closely with each other during the integration and testing phase. All problems and conflicts were constructive and helped to improve our project and relationship with each other. Our friendships have strengthened after the project.

5 Project Contribution and Experience

Austen Chan - Chief Technical Officer (CTO)

I was responsible in all the circuit designing and implementation. By considering the system requirements of our project and budget, I searched for the most suitable IR sensor and accelerometer. Since the original control board of the remote control vehicle would no longer be used, I designed our own control board which was composed with the Gumstix minicomputer, digitalization circuit with filtering, pull-up network, motor control circuit and LED indicator circuit. After designing the control board, I populated it on a prototype circuit board and routed the connections with wires. Due to the space limitation of the vehicle and prototype board, I placed all the essential components on the board before soldering them so that I could adjust the components' location for smaller space occupation. I was also responsible in debugging the circuit and looking for solutions when the circuit did not work as we expected.
From the last 13 weeks, I learned how to select components by considering the system requirement, budget and timeline. Considering timeline was important because some selected components might be out of stock and I had to look for substitution components. When I was implementing the circuit, I learned to pay more attention to detail because any routing error could lead to damage of the circuit and components. I also learned to be patient because everyone was lining up for the soldering iron. Finally, I gained practical experience in electronics circuit design in additional to the theories that I learned from school. In the non-technical side, I learned the team dynamic. A team was always better than a single man because no one could be specialist in all area, but a team could be. I also learned how to do a better time management for this project, other academic courses and personal life.

Brian Cheung - Production Director

It has been four months and it allowed me to spend more than 40 hours each week to accomplish this project. I have learned great many things from this project. I have participated since the early stage of the project, from initiating to development, from development to integrating and testing. I provided ideas of whether the project is feasible and achievable within our time strain and budget. I also helped set up Gantt chart to ensure our progress keeps up with schedule.

For team dynamics, it is great to work with people who you already knew or worked with previously. I trust my teammates that they are able to deliver quality work in a timely fashion. I can always rely on them with works that were assigned to them. We constantly exchange most updated information with each other and help each other with problems encountered. Our relationships with each other have improved after the project.

Working with Rongen, we setup the software environments and drivers that are necessary to develop our software on our microcontroller. I learned about the Linux environment and OpenEmbedded cross-compiler. I also set up Bluetooth connections with the PS3 controller as well as the six-axis motion sensing mechanism to work with our microcontroller. In addition, I setup the General Purpose Input Output (GPIO) pins that need to be used with our sensors and motors. Not all GPIO pins can be used because most of them are pre-defined by the controller to perform a certain function or as input/output pins for other hardware parts. I provided sample working codes that can be used to program the GPIO pins and perform several test procedures on the state and usability of the GPIO pins.

Working with Bruce, I provided ideas on our AI algorithm in automated mode. I discovered some bugs the algorithm and provided solutions to the problem. In addition, in order to test our AI algorithm in automated mode, I designed the maze for testing and analysis with WingKit. The maze can also be used during demonstration of our project to prove the workability of our AI algorithm.

Unfortunately, I did not involve in much of the work that was done with the electronic components implementations due to time and energy limitations. However, I do have understandings of what are being done with the components in order to have the desire functionalities.

Overall, I enjoy doing the project because I gained knowledge in all different aspects of a project development life cycle. The project gave me the opportunity to extend my knowledge in this field. I greatly appreciate this project opportunity.

Bruce Wong - Manufacturing Director

I have completed the entire programming aspect in this project. After such a long period of programming, my programming skills have increased quite a bit. During this project I have learned how to decode and information from a PS3 controller using Bluetooth. I have also learned how to program GPIO pins, and learned to create my own automation algorithm with help from my team members. At last, I have learned how to implement a tracking algorithm of a moving object.

Rongen Cheng - Chief Executive Officer (CEO)

"This is chaotic!" Nothing can described what we have gone through this four months madness. Applying my previous experience on project management, it is giving me a good indication on what problems we will going to face. Unfortunately, experience is only a better synonym to history which does not foresee the future. As five of us are having other academic works, projects in the mean time, time management throughout the project is harder than ever. Luckily, we all devoted into this project which we sacrifice our free time to make this happens.

During week 6, we hit a wall which we have had a moment that we want to redefine our project requirements. This wall made us had a feeling that we were not able to complete our initial project requirements as using the controller to control the vehicle. Fortunately, all of a sudden, we are able to decode the transmitted signal from the processor. Things are getting straight prior this point but this is not the major milestone of this project.

At the night of March 28, 2009, we were trying to integrate the subsystems together as we planned earlier. Suddenly, a smell of burnt plastic arose. Next thing we discovered was a calamitous incident. Our processor was burnt, it was a processor which took three weeks to ship previously. At this time, the remaining days before demo day were exactly 21 days. We were totally lost our hope on getting a new one on time. One week later, miracle happened. We received our new processor. Since I have backup all the previous work on the Linux Kernel environment system setup and procedures, I was able to setup the new processor with all the required drivers and modules within one day. After some more testing with the previously worked code, we were able to steer back on track with our project.

Now, we held a meeting on how to prevent this kind of accident happens in the future. The result was to additionally include a circuitry protection towards the processor. Thes rest is history. The most important thing keep us working closely together is that we take all blames as a driving force to make our work perfect. This is something that teaches us after this four months madness.

Wingkit Lee - Chief Financial Officer (CFO)

As a system engineer with more knowledge in the mechanical aspect than others in my team, and the fact that I worked as a assistant mechanical engineer as a Co-op, my main task was to be in charge of all the issues and problems mechanically. Although I helped out in some soldering and software programming, most of my time was dedicated on the mechanics of the remote controlled car. There are two types of mechanical work that I did, ones that were originally planned and those that came up when problems arise as the project progresses. On the hand, I was heavily involved in the documentation of the project as I wrote some of the contents and was the editor for the group.

The tasks involved in the original plan includes those that were involved with the design and construction of the top and bottom platforms, heat-sinks for H-Bridges, casing for the controller package, and the overall structure mechanism. I was involved in every mechanical aspect, from designing what screws and materials to use to constructing it to putting it all together. For the work involved in fixing problems, I had to design new ways or modify some of the mechanisms. Some of the major ones include designs to reduce the overall vehicle weight, the elastic band and the spring mechanism for returning the wheels straight, the reduction of weight distribution on the wheels, motor casing refinement, and to match changing needs and design in other aspects in the project.

In this course, I learned lots in terms of technical skills and interpersonal/social skills. On the technical side, I gained vast experience in hands on Tech-Ed skills and in developing new designs or working around with limited resources to achieve tasks and fix problems.

On the side, I learned how Bluetooth connections work with the GumStix, and gained experience in programming in Linux environment. Moreover, my knowledge in the basics of electronics enhanced when I assisted my group mates in their field, such as the functions of H-Bridges, soldering skills, and electronic signalling. Lastly, working with a group of 5 with all of them as friends before this semester taught me how to work with people you know in the same project. Our backgrounds, knowledge, and experiences are diverse and we had to accept each others' ideas and respect each other. Another noteworthy point is how to put up with some unlikable behaviors from friends and avoid any serious conflicts. After all, the group atmosphere was great and everything worked out fine.

6 Conclusion

6.1 Future Work

In terms of hardware, we would like to replace our infrared (IR) sensors with ultra-sound sensors. Ultra-sound sensors can increase our sensing sensitivity as well as wider detection angle. In this case, blind spots can be reduced or even eliminated. Moreover, we would like to replace the 4 wheels with caterpillar track. This allows the vehicle to a very small turning radius (almost none). This can improve the path mapping algorithm and the vehicle can become more versatile. A GPS module can be added to expand our functionality to point to point navigation.

Considering software aspects, we can add an extra layer on the LCD display to additional information regarding the status of the vehicle, such as direction of travel and speed. We can also use our existing camera for image processing. This can enhance our path recognition and obstacle recognition. With these improvements, a more enhanced selection algorithm can be implemented.

6.2 Recommended Improvements

6.2.1 Components Selection

The IR sensor could be replaced with ultra-sonic sensor because ultra-sonic sensor has a better sensing range and accuracy. The main problem of using IR sensor on our project was we had to use a large amount of IR sensors to gain better accuracy on obstacle detection. However, ultra-sonic sensor has the sensing accuracy advantage over IR sensor. More IR sensors would also lead to higher power consumption. Using ultra-sonic sensor could help solving the power consumption problem because we only need to use few of them. Opto-isolator could be use as pull-up network instead of comparator to gain further protection to the processor. If possible, the Gumstix minicomputer could be replaced with other micro-controller unit with Bluetooth module. Gumstix had limited sources on documentation, and it did not have an analog to digital converter. With an ADC, we could also gain better accuracy on obstacle sensing because the processor could know how far was the obstacle instead of knowing the obstacle was within sensing range.

Moreover, the GPIO connector pins of Gumstix were so small and hard to solder. Other choices of micro-controller might have advantage on connector.

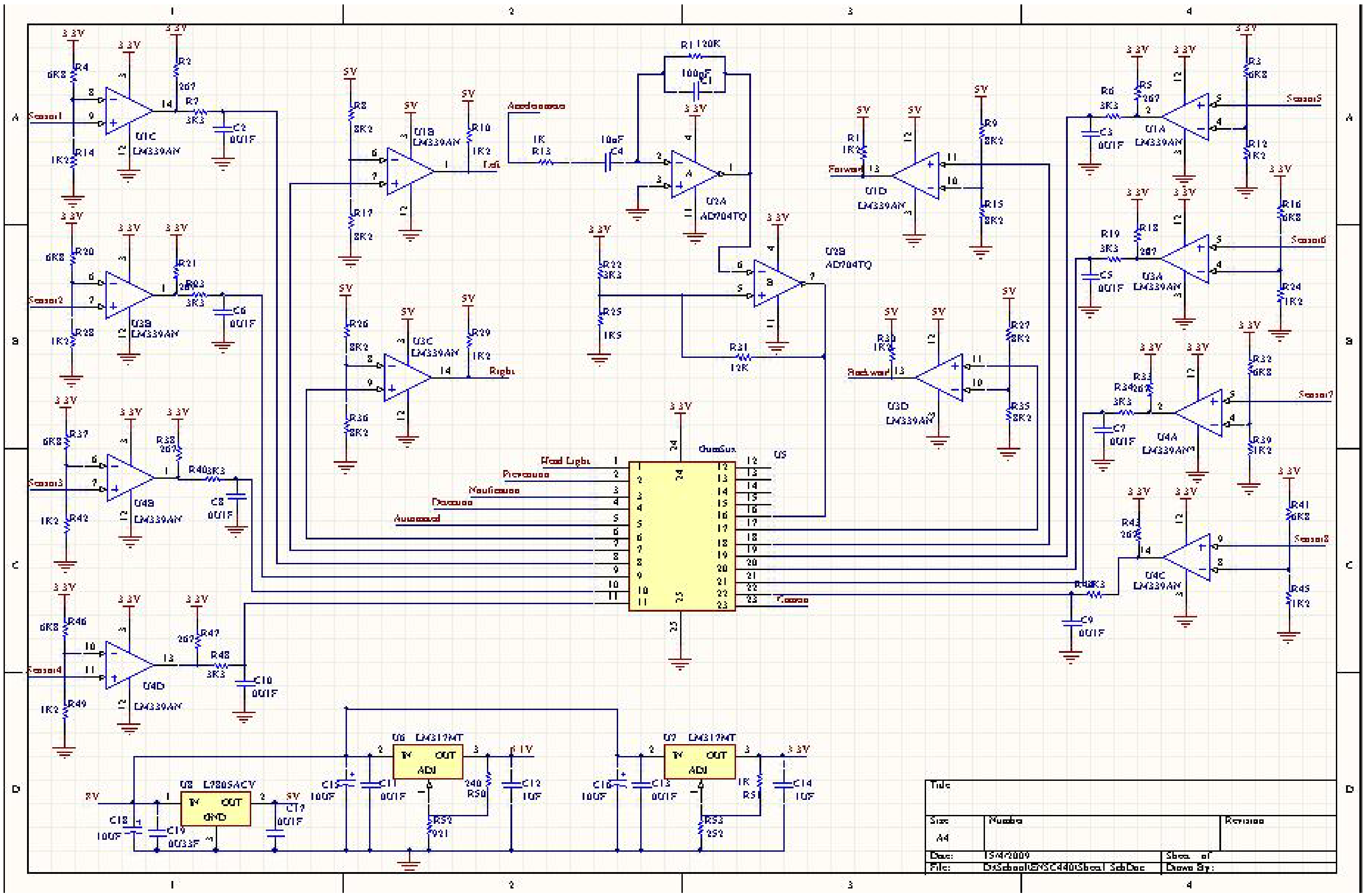
6.2.2 Signal Handling Improvement

The calculation of the turning angle may be improved by taking consider of the pitch rate. The automation of the vehicle can be done indefinitely until obstacles detect instead of checking and output a signal every clock cycle. That is it can send a long signal when automation is needed instead of an impulse to improve the efficiency of obstacle avoidance. Lastly, the speed and turning power of the motors can be fined tune by changed the output pulse frequency and duty cycle to achieve a better response during manual and automation mode.

6.2.3 Mechanical Components

The final version of the remote vehicle prototype uses the basic backbone components of the original car, such as the chassis, acceleration motors, and wheels. The vehicle also has a top and a bottom platforms attached to it. On the top plexiglass platform, IR sensors are mounted on the edges and a web camera is mounted near the front of the platform. The final solution to the weight problem uses a plastic band to secure the wheel chassis position. For the wheel rotation problem, a spring mechanism is implemented to return the wheels back to the forward position by the spring's force.

7 Appendix - Overall System Circuitry (Page 1)



8 Appendix - Overall System Circuitry (Page 2)