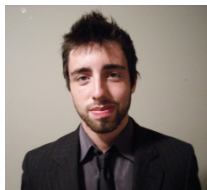


About the Design Team



Aaron MacNeill is a third year electrical engineering student at Dalhousie University. His home is based out of Labrador, and plans on becoming a power systems engineer following the completion of his undergraduate degree. Aaron specialized in the hardware & design sector of **Project Gattaca**.



Lance Taylor is also a third year electrical engineering student at Dalhousie University, and his home is based out of New Brunswick. Upon completion of his undergraduate degree, Lance plans on attaining his Masters in Business Administration. Lance specialized in the software & control systems.

Acknowledgements

The design team would like to acknowledge the following individuals for their assistance in the development of the project:

Dr. Peter Gregson, Dalhousie University
Paul Dexter, Dalhousie University
Jared Smith, Dalhousie University
Mark Leblanc, Technician - Dalhousie University
Peter MacDonald, Dalhousie University
Francois Chaisson, Dalhousie University
Chris Hill, Technician - Dalhousie University
Ian McKenzie, Technician - Dalhousie University

The above individuals provided technical assistance as well as facilitation for the development and design of **Project Gattaca**.

Additional thanks are directed toward the following organizations:



Executive Summary

Project Gattaca is a multi-sensor driven robotic navigational platform designed in fulfillment of the ECED 3901 Design Methods II class at Dalhousie University. Project Gattaca is a fully autonomous navigation control system that attaches to a supplied mechanical base. The goal of the project was to design a robot that could navigate successfully through a series of obstacles that are designed to represent the ledges off the coast of Halifax, Nova Scotia. These obstacles are placed strategically on a competition playing field, with a button at the finish location. Through making use of a provided robotic parts kit that included basic analog and digital electronic components, sensors, motor drivers and a microcontroller could be constructed to control this robotic base, directing it through the course.

This project was initiated in May of 2009, and progressed throughout the following months until the competition date on July 23, 2009.

The strategy selected by the Project Gattaca design team was to design a functional robot, while minimizing complexity, and maximizing efficiency and versatility. Out of the various possible movement strategies available on the competition playing field, it was deemed most technically feasible to proceed along the inner edge of the field, while using a reflective tape sensor to avoid falling off the competition surface. The motor driver selected for the project was a half-bridge bi-directional driver, which would allow sharper turns, and the versatility of being able to move in reverse. The sensors needed, as previously stated, was a reflective tape sensor that used a photo-sensor and a high intensity light emitting diode, as well as two tick-wheel sensors that detected the revolutions of the robot base wheels. By utilizing these sensors, the robot would be able to slowly and accurately navigate its way into the 4-point zone consistently.

The completed project did not resemble the original planned design, as several complications arose during the construction & testing phase. More detail is provided in the report on this matter. The tick wheel sensors and their associated control circuitry worked as expected, as well as the reflective tape sensor. The programming system developed in the C+ programming language provided adequate control of the robot, and the supplemented motor driver (see report) also performed to par.

The robot attained second place in the competition, and was able to successfully hit the button on a number of trial runs.

Table of Contents

About the Design Team.....	1
Acknowledgements.....	1
Executive Summary.....	2
1.0 Introduction.....	4
1.1 Operation of Final Robot.....	4
1.2 Project Phases.....	5
1.2.1 Design Phase.....	5
1.2.2 Construction Phase.....	5
1.2.3 Testing & Revision Phase.....	5
1.3 Report Sequence.....	5
2.0 Discussion.....	6
2.1 Work Completed.....	6
2.1.1 Motor Driver.....	6
2.1.2 Time Delay Circuitry.....	6
2.1.3 Wheel Tick Sensors.....	7
2.1.4 Reflective Tape Sensor.....	7
2.1.5 Programming and Control Systems.....	8
2.2 Problems Encountered.....	8
2.2.1 Motor Driver Issues.....	8
2.2.2 Wheel Tick Sensor Issues.....	9
2.2.3 Game-Day Explosion.....	9
2.3 Performance of Completed Solution.....	9
3.0 Conclusion.....	10
3.1 Evaluation of Progress.....	10
3.2 Statement of Success.....	10
3.3 Recommendations & Changes.....	10

1.0 Introduction

In the initial design phases of **Project Gattaca**, it was decided that the mission statement would be *to create a fully functioning robot, which utilizes the utmost simplicity in design, while maximizing efficiency and performance*. It was understood that selecting a simple design would in fact reduce the final versatility of the robot, yet it was accepted that by choosing such a path, it would also reduce the chance of technical failure due to over-complexity.

It should also be stated that within this report, there will be references made to the project design, and the final product. These two entities are not one in the same, as certain unforeseen events transpired that induced a sudden change in design, in order to ensure adequate function within the competition.

1.1 Operation of Final Robot

Although this will be described in more detail in the discussion section, the final robot function is provided here in two stages – the final robot product, and the final robot design.

The final physical robot (as of the competition day) consisted of two tick wheel sensors, a uni-directional motor driver, and a simple dead reckoning programming algorithm. The performance of the robot depended on its ability to drive in a straight line (as it was decided not to include the alignment algorithm). This was accomplished by simply pulse-width modulating the slower wheel at a lower maximum value than the other. This ratio was determined through a brief test run once the robot base had been presented. The robot would then travel for a pre-calculated number of ticks, turn one wheel for another sequence of wheel-ticks, proceed in a straight line, turn once again, and then finally drive towards the button. On competition day, several adjustments were made to this process, and these are again outlined in the discussion section.

The operation of the robot *design* which was not ultimately implemented, consisted of two tick-wheel sensors, a reflective tape sensor, a half bridge bi-directional motor driver, the Atmel processor, and a time delay circuit. The objective of this design was to drive along the inner edge of the surface, while sensing the reflective tape and driving into the four-point region. Up until the day before the competition, it was understood that only a part of the robot needed to touch the point zone to receive those points. After clarifying that this was not the case, the design merely needed to be adjusted to move the castors entirely into the four-point zone. Although this design was not used on the competition day, it did provide a safer and more confident route to the button, as no interference from the other robot would affect this route.

Project Gattaca also included a backup algorithm that ensured a quick route to the 3 point zone for the first round. This allowed the robot to simply drive straight ahead, make a quick left turn towards the

ledges, and stop in the zone. By doing this, it ensured that the team would move on to later rounds, where more complex strategies could be deployed.

1.2 Project Phases

The project design process was broken into several phases, as shown in the Gantt Chart in Appendix C. When laying out the project phases, a number of project management techniques were utilized (as discussed in the course). The SCRUM design method, combined with the Stage-Gate method, were key factors in outlining the project timeline.

1.2.1 Design Phase

Within this phase, the designs for the motor driver, sensors, and control circuitry were drafted and tested. Also, preliminary programming test programs were designed and tested. This phase allowed for the identification and elimination of “project killers”, or elements that would introduce problems down the road. Also, it allowed the project team to compile a portfolio of designs that may not be used in the final design; however, they could be called upon at any time, with the assurance of their performance.

1.2.2 Construction Phase

Once the designs had been completed and selected, the project moved into the construction phase, where all circuits were built, and sensors were constructed. The programming did not have a large part in this phase.

1.2.3 Testing & Revision Phase

During this phase, the design team compiled all the constructed elements and assembled the code and began running through a series of tests. It was during this phase when most issues arose, and an iterative process was used to revise and re-test the design.

1.3 Report Sequence

The above phases will be alluded to in the following discussion section, and each component will be broken down and explained in detail. The information that follows is based on the robot design, not the final product. There will be a section explaining the issues that arose from the testing phase, and the last minute adjustments that were made.

The motor driver, sensors, programming, and delay circuitry will be explained in the work completed section, and the performance analysis will follow the problems and issues section.

2.0 Discussion

The following information details the technical research and development employed by the Project Gattaca design team, along with providing a detailed summary of problems encountered and an analysis of performance.

2.1 Work Completed

Although all the following completed work was not included in the final robot design, it still remained part of **Project Gattaca** and has been noted below. The motor driver, delay circuitry, wheel-tick sensor, reflective-tape sensor, and programming algorithms will be discussed.

2.1.1 Motor Driver

As stated above, after planning out the proposed strategy for the chosen movement path, it was deemed necessary to build a bi-directional motor driver. For the needs of this specific strategy, only 12V was needed to power the motors, as speed was not overly essential. With this in mind, the design proceeded by drafting up a motor driver design, and testing it in every possible situation. When designing the driver, it was known that on startup, the motors could draw up to 10A of current, meaning the NMOS and PMOS transistors must be biased at 5V. Adding a safety buffer of +2V, the transistors were biased at 7V. Given this constraint and the constraint that the microprocessor can only output a - 5V signal it was evident that more circuitry would be needed between the microprocessor and the mosfets. See Appendix A Figure 1 for the design of this circuitry.

This motor driver circuit was tested with the small motors during a lab session, to prove its operation. Originally, BJT transistors were used with the signal from the microprocessor, in an attempt to amplify the 5V (or approximately 4.6V) output signal. After testing the motor driver with a test signal of 5V, it was found that the operation of the motor driver was adequate. The driver was then connected to a robot base, and driven with a series of physical switches, to test for blocked rotor current conditions and start-up currents. After these tests, it was proven that the motor driver worked correctly and adequately.

2.1.2 Time Delay Circuitry

A time delay circuit was later incorporated into the motor driver, in an attempt to delay power supply to the motors to eliminate all start-up transients of the electrical devices - mostly concerned about the microprocessor and voltage regulator transients. This time delay circuit was designed so that it would work with the pre-existing motor driver. See Appendix A Figure 2 for the design of the time delay circuit. The choice of resistor and capacitor values for the low pass filter was tailored to yield a 2V charge for the capacitor after approximately 80ms. The figure of 80ms was obtained from knowing the start up time of the microprocessor and the voltage regulator, which were far under 80ms. The value of 2V was derived from knowing the mosfets are not biased until it has a gate to source voltage of 2V; therefore, if it takes 80ms to charge to 2V then it is guaranteed that the transistor will not be on for 80ms. The biasing of this

transistor is what will affect whether the transistor that is to supply power to the motor driver is biased properly or not. This time delay circuit was tested thoroughly before it was built into the motor driver. Tests that were performed on it were as follows: the voltage across the capacitor was checked to see that it did reach a 2V charge after approximately 80ms; the mosfet was tested to see that it did not turn on until a larger than 2V signal was applied to the gate; the voltage at the drain of this mosfet was monitored to see that on start up the voltage was at the same voltage as the source of the power mosfet, and that after a certain time this voltage went to zero. In the end the time delay ended up being approximately 130ms and it could be seen on start up of the motor driver while testing it. With the time delay circuit there was a small, but noticeable time delay between switching on the power and the wheels beginning to turn.

2.1.3 Wheel Tick Sensors

The tick sensors enabled the microprocessor to detect how far a specific wheel had turned, by monitoring the number of transitions from black to white on the wheel tick indicator. See Appendix A Figure 3 for the design of these sensors. It was difficult to ensure the accuracy of results obtained from these sensors. From a circuitry level, it was important to eliminate noisy pulses, to ensure that no additional "ticks" were being counted. From a hardware/mechanical perspective, the starting and stopping of the wheels presented sporadic results. The latter of these two issues was accounted for in the programming, by creating a minor delay prior to reading in any ticks, and resetting the tick counter after starting the motors (and another small delay). The first issue, however, of the noisy high frequency components was eliminated through the use of the circuit above, and the careful selection of capacitance values.

The testing of this circuit took place after the motor driver was fully operational with the robot base. This was because it is very difficult to simulate the rotating tick wheel without an actual spinning one. During this testing it was seen that this circuit provided a very reliable signal for counting the number of ticks.

2.1.4 Reflective Tape Sensor

The reflective edge tape sensor was built to detect the boundary of the obstacle course, and would prevent the robot from going off the surface and potentially losing points. See Appendix A Figure 4 for the design of this sensor. This detector was built from a phototransistor and a LED that were aimed such that reflected light from a surface perpendicular to the LED would reflect into the phototransistors. This would allow the phototransistor to conduct, sending a signal to the microprocessor which would then deal with the sensor accordingly. This sensor was designed, built and tested but there was not enough time to implement it on the actual robot. The testing that was done was to position the LED so that reflection occurred on a reflective perpendicular surface to the sensor. A shroud was made around this sensor to protect it from random ambient light interfering with the sensor. This shroud worked well in keeping out other ambient light. The signal that was generated from the sensor varied between 0.5V and 4.9V. The signal would then be fed directly to the microprocessor Analog to Digital converter (see next section) where it could then be manipulated to ensure accurate readings.

2.1.5 Programming and Control Systems

The provided processor, Atmel ATMEGA8535, was programmed using C programming, which was downloaded to the microcontroller board through a USB device. The programming was designed using an object oriented development system, allowing functions to be called throughout each module. The control system was broken down into several parts: the main control system, the Analog to Digital converter module, the pulse width modulation module, the drive function module, the timer module, the tick sensor module, and the metal sensor module. The source code has been attached in the appendix, with some of the comments stripped for space reasons.

The inputs to the microcontroller consisted of the tick-sensor outputs, the reflective edge sensor, and the turning radius control pins (explained later). The outputs were the pulse-width modulation pins, and the direction control pins.

The programming operated on a polling basis, where the inputs were tested once every 0.940 ms. At each interrupt, the tick sensors were read, the ADC input was read, and the timer counter was incremented.

It is difficult to explain the function of all of the code in the appendix; however, a specific element of the code that should be explained is the alignment algorithm. To ensure that the robot was running straight at run-time, the motor was allowed to drive straight for 15 ticks, then calculate the ratio between tick sensors, create a ratio of motor speeds, adjust the other motor accordingly which calibrates the speed of the motor, then re-align the robot. This algorithm worked especially well, and was documented on video. Prior to this method, an algorithm was tested that changed the PWM at runtime, without using the "Drive" function (see source code). When doing this, the microcontroller appeared to go into an infinite loop that caused the motor driver to overheat. Therefore, the secondary method was chosen to ensure no unexpected explosions, at the sacrifice of speed.

2.2 Problems Encountered

As is the case with most technical projects, errors and issues arise in the most unexpected places. Project Gattaca was not immune to this tendency, as the design team experienced more issues than could ever be expected.

2.2.1 Motor Driver Issues

During the testing & revision phases of the project, the motor driver failed on a number of occasions. Extensive research into the cause of these faults was performed, and no conclusive evidence was found as a source of this issue. Originally it was thought that the delay circuitry was malfunctioning – however it was proven that this was not the case. It was then thought that the microprocessor was sending ghost signals to the controller. Finally, tests were performed to identify any unpredicted transients of the components. Unfortunately no conclusive results were found.

Therefore, as a last ditch effort to develop a working motor driver for the competition day, the team switched to a simple uni-directional driver. This has been referred to previously as the final *product*, but does not represent the **Project Gattaca** design. This design simply was a mosfet with a flyback diode across the motor, and a biasing voltage of 8 V, with the voltage division of a 12 volt output signal from opto-isolators. The programming remained the same, however some minor adjustments needed to be made. The strategy was then changed to pursue the outer route. This choice presented a longer route, but the unidirectional driver did add a significant speed boost. By using this method, the reflective tape sensor was not needed, nor was the ADC module.

2.2.2 Wheel Tick Sensor Issues

On competition day, one of the tick sensors was destroyed due to a short circuit across the phototransistor. Therefore, only one sensor was used. This was accomplished by switching the sensor from wheel to wheel using Velcro strips. Since only one wheel was turning in a specific heat, this issue did not present much hindrance. However, by only having one sensor, it was difficult to align the speeds of the wheels. Therefore, a series of jumpers were attached to Port B of the microcontroller that allowed the operator to adjust the turning radius by adding or removing numbers of ticks to the turning radius. This could be done at runtime of the competition, which allowed the team to fine-tune the behaviour of the robot.

2.2.3 Game-Day Explosion

Due to an unexpected mental error on behalf of one of the team members, the robot capsized on itself during a trial run, and the motor driver exploded. However, the design team, with the help of one of the technicians, was able to repair the motor driver. Following this explosion, the robot hit the button directly on its first trial run.

2.3 Performance of Completed Solution

Although the completed solution did not represent the final design, it did perform remarkably well. The robot hit the lighthouse button on a number of occasions. Although the robot did not hit the button in competition, it performed much better than expected. The team was satisfied with the performance of the final product; however, it was unfortunate that the final design did not have a chance to be represented in the competition.

The final product, as stated above, used the unidirectional motor driver to drive straight for a number of ticks, turn towards the middle, drive into the center of the competition surface, and then turn into the final stretch towards the button. Using only one tick sensor and simple programming, the robot met the conditions of the Project Gattaca mission statement (see section 1.0) by maximizing simplicity and efficiency.

3.0 Conclusion

3.1 Evaluation of Progress

The progress of Project Gattaca during the design phases adhered to the proposed project management schedule relatively closely. The team fell slightly behind schedule due to the issues that arose with the motor driver, but this was accounted for with an additional surge of hours put into the project several days prior to the competition. The design phase took slightly longer than indicated in the Gantt Chart.

3.2 Statement of Success

In terms of the ultimate goal of the competition, **Project Gattaca** out performed every robot in terms of ability to hit the button. After the final round, the robot was driven one more time, and hit the button head on. Unfortunately, lighting of the gym affected the final turn during the final round, which would have otherwise ensured a victory in the competition. Overall, the design team is happy with the success of the robot product. The team also feels the the robot *design* – which was not used in the competition – would have also proved to be just as successful. Ultimately, **Project Gattaca** yielded great results, and is a successful project.

3.3 Recommendations & Changes

It is recommended that if the project was to be re-done, more time would be allocated to the design and testing of the motor driver, and the entire design process should have started several weeks earlier. Insufficient time was allocated to testing the programming with the motor driver, as complications arose very late in the design stages. This made it difficult to assemble all the components and test accordingly. Therefore, better communication between team members, combined with an earlier start to the design work will ensure a more successful implementation of product design.

Judging by the performance of other robots, simplicity overtook complexity – however, in terms of experience and lessons learned, by choosing a more complex route, and challenging one's self, more information is learned, and the objective of the *course* – not the competition – is successfully attained.

It is also recommended that if further work is to be done on Project Gattaca, the half-bridge motor driver be re-implemented into the design, and further testing be done with the reflective edge sensor.