2013 Third Professional Year Project

Drive-by-wire Go-kart

Nissanka Weerekoon

9/16/2013

Supervisor: Dr Andrew Bainbridge-Smith

ABSTRACT

This project is the continuation of prior years' work towards developing an autonomous go-kart. It provides a way to be part of the global push towards autonomous vehicle navigation on a small scale. In 2011 a number of Controller Area Network Bus (CANBus) compatible PCB boards were developed for controlling the various functional modules of an electric go-kart and appropriate actuators were chosen. In 2012 a high level Application Programming Interface (API) was written in Python to control the various modules from a laptop on board the go-kart, however was never tested on the go-kart. This year the project had focussed on getting the CANBus control system integrated with the go-kart and getting the go-kart moving in a laptop controllable state. This was achieved successfully with the go-kart mounted off the ground and capable of being controlled via a Graphical User Interface (GUI). Moving the go-kart around on the ground would be a suitable next step. In addition, the Python based control system was integrated with a checkerboard tracking algorithm in C++ to demonstrate its autonomous response capability. This was successfully able to adjust motor speed and steering at a basic level however would require improvement for real outdoor tracking. The progress made this year provides a working platform upon which further developments in autonomous navigation, computer vision, safety systems and CANBus control system design can be made in parallel.

1. INTRODUCTION

The goal of this project is to integrate an existing CANBus drive-by-wire system onto an electrically powered go-kart, providing basic drive-by-wire movement and control of all features. This then allows the detailed development of autonomous movement and computer vision systems on an existing platform in future years.

Autonomous control is a modern technology becoming ever more popular in the engineering world. Autonomous control of ground vehicles in particular is rapidly developing with the push towards driverless motor vehicles for both passengers and goods. Driverless control or even computer aided driver assistance has the potential to be greatly advantageous in terms of public safety, reduced fuel costs, reduced emissions, reduced time costs and more efficient transportation in general. Significant developments are yet to be made in the autonomous vehicle technology itself as well as the infrastructure and policies required for the public integration of driverless cars. Events such as the DARPA Urban Challenge provide developers a chance to test their vehicles in mock situations [1]. The motivation for this project comes from this global push towards autonomous movement and navigation. As inexpensive autonomous vehicles would be highly advantageous, this project focuses on a smaller, less complex scale, incorporating a go-kart chassis.

With only a small number of developers usually working on the project each year and the complexity of developing an autonomous vehicle, the end goal of autonomous navigation must be achieved over a number of years. Each year partial progress is made towards the overall autonomous navigation goal in the form of a final year project in the Electrical Engineering Department. To date, the project is a continuation of prior work over 2 years. In the first year of the project, 2011, one of the Department's electric go-karts was modified to run via actuators and a set of PCB control boards were developed. In 2012 an Application Programming Interface (API) was written to enable high level control of the go-kart movements. However this was never tested on the go-kart. This year it was decided to focus on integrating the entire existing system with the go-kart platform and use the API to control the basic movements.

When work on the project began this year, a lot of time was spent on research and familiarisation of the work done thus far. This included testing and clear identification of areas needing work. This lead to further revision of the project goal for this year. The primary focus would now be on getting the go-kart moving and controllable via a laptop. Autonomous movement would only be added in the form of steering, motor and brake responses to a computer vision target tracking algorithm. This report goes on to outline the problems encountered while integrating the control system with the go-kart and the solutions that were developed. It also highlights the difficulties in beginning work on a project where work has previously been completed.

2. BACKGROUND

2.1 Project History and Evaluation of previous work

The project was first started in 2011, with the intention of transforming one of the University of Canterbury Electrical and Computer Engineering Department's go-karts, pictured in Figure 1, into an autonomous vehicle. A team of four undertook the project and found it to be more work that appropriate for one final year project. The team therefore limited their work to choosing the steering and brake actuators and designing seven PCBs [2].

Figure 1: One of the Department's go-kart prior to integrating the CANBus system

Five of the PCBs were for the CANBus control of the brake, steering, motor and sensor modules through a communications module connected to a laptop. These PCBs were all given the same layout to ease both development and manufacturing; however each board was only populated with components required for its individual function. All boards use AT91SAM7XC processors. There were several mistakes made in the designs however work-arounds were successfully implemented when populating the boards. Two motor driver PCBs were also designed for driving the steering and brake actuators. These were found to be inadequate and replaced the following year. The go-kart chassis was also modified by this team for the integration of the CANBus system with the required actuator attachments. The brake actuator bracket had been misplaced and needed to be redesigned and manufactured this year.

The 2011 team also looked into safety systems, software development practices and computer vision target tracking for the go-kart [3], [4], [5]. Software development identified the use of unit tests and continuous integration; however the implementation of these methods in the project appeared minimal at most.

In 2012 work on the project was continued by a single student, similar to this year. The software side of the project developed in 2011 was deemed unsatisfactory for progression of the project and developing a new high-level software control API for the go-kart became the primary objective for that year. A bottom-up approach was implemented to accomplish the required control levels: peripheral control on each board; inter-board CANBus communication; laptop control via USB. All these levels of control were achieved; however the system was only tested on bench-top and not tested on or calibrated for the go-kart platform [6].

Upon receiving the project this year, initial inspection highlighted a number of uncompleted tasks from 2012. These were the inadequate number of unit tests, need for new board connectors and implementation of motor module communication with the PWM student board.

2.2 Development Platforms

Software development was done mainly under Linux as the existing code, debugging and makefile formats and procedures most easily supported this. Documentation from 2012 briefly outlined a method for setting up the compiler and operating the system through a command line and python script combination under Linux; another reason why it was used. As the project this year would require development and debugging around the go-kart situated in the machines lab, a department desktop Linux machine could not be used and a decision was made for development and running of the go-kart to be done via the developer's personal laptop. The 2012 documentation was however now invalid or lacking in some areas therefore procedures for installing Linux on a development laptop partition along with setting up the ARM compiler had to be researched. The procedure and links to instructions referenced were documented for ease of future setup [7].

3. DESIGN PROCESS

3.1 Existing system functionality

The first step towards integrating the previously developed system onto the go-kart was to identify what functionality was already present and confirm its working condition. Much time was spent during the early stages of the project becoming familiar with work done over each of the previous years and what parts had been changed from year to year. As an initial test, the CANBus system with actuators was assembled on bench as was done in 2012 and the results of the previous year were recreated. The setup was as shown in Figure 2 with the motor board essentially having no functionality as there was no PWM student board connected.

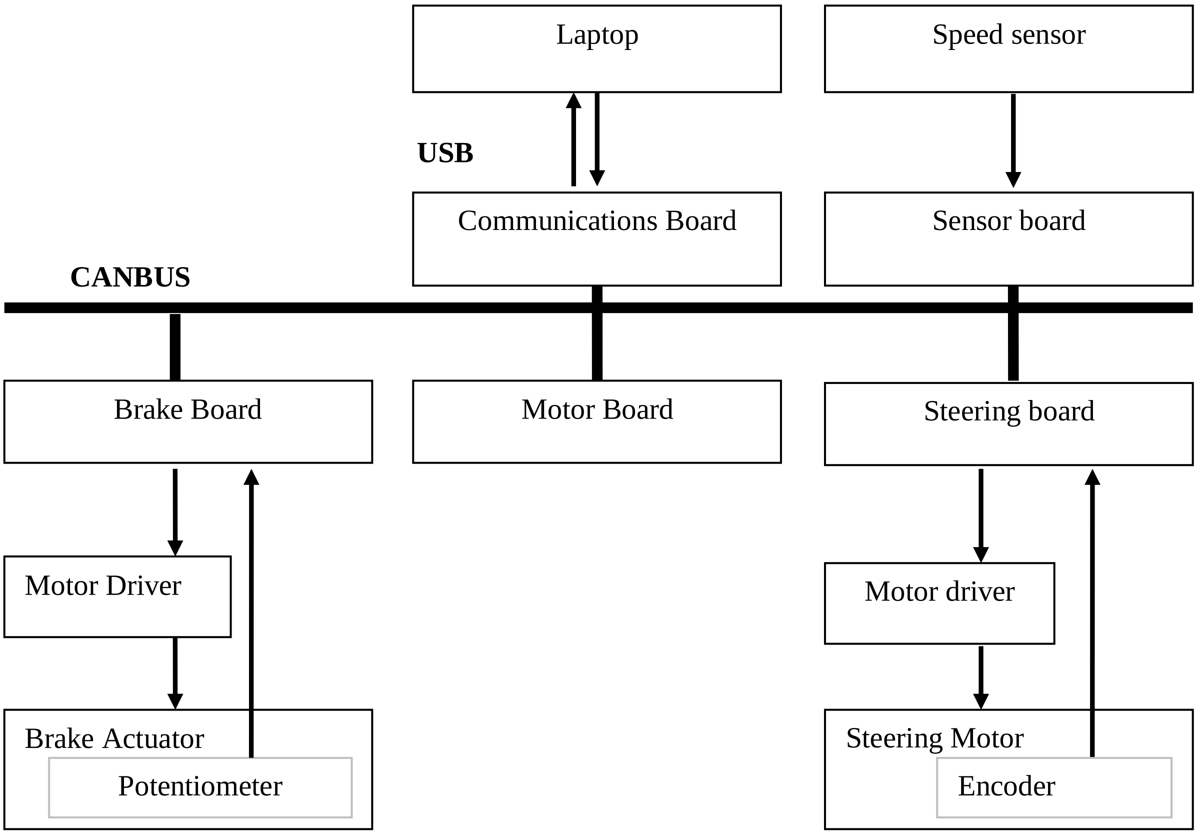

Figure 2: CANBus control system block diagram upon receiving project

The system consisted of the 5 boards: communications (comms); motor; steering; brake; and sensor. Both the brake and steering board connected to motor driver boards purchased in 2012, which in turn connected to the brake and steering actuators respectively. Both actuators gave positional feedback signals via potentiometer or rotary encoder which were passed back to the brake and steering boards directly. The steering board also connected to two limit switches used in the calibration process and incase of a steering malfunction. The sensor board connected to an inductive proximity sensor intended to measure passing teeth on the rear axle sprocket for calculating speed. The motor board was left unconnected at this stage as the PWM student board interface was yet to be implemented fully. The comms board was connected to the laptop via mini USB cable. All the boards were connected to each other via CANBus configuration using CAT5 ethernet cable. The boards were powered on bench using a 24V DC supply, which was stepped down to 5V on the brake board and distributed via the ethernet cables. The actuators were powered using a separate 24V bench supply as a combined startup current draw of approximately 1.2A was found to be required.

Upon power-up the system followed the expected calibration routines, as outlined in documentation from the previous year. A number of python unit tests could be run from the laptop command line which successfully cycled through possible commands for the brake and steering modules. The motor module passed its unit tests even though no PWM student board required to interface with the motor was connected. The speed sensor had to be powered from the 12V regulator on the brake board as the requlator on the speed board was malfunctioning. Speed module unit tests were runnable however accuracy and success in these was impossible to determine without being connected to the rear axle sprocket of the go-kart. As the comms board only acted as the CANBus mediator it appeared to function correctly, passing messages back and forth from the CANBus to USB interface.

In terms of mounting the system on the go-kart, provision had already been made to attach the steering actuator in place of the steering wheel. However the brake actuator mounting bracket was found to be missing. Provision was also found to attach the steering limit switches around the steering column and the inductive proximity sensor near the real axle sprocket. The boards were placed around the go-kart boardy near their respective control areas however there was no means to secure them. There was also no known connection for powering the boards or actuators from the go-kart battery supply.

This initial step of recreating previous years' results and confirming the state of all components of the system provided insight into possible work that would need to be done. This, along with a comprehensive 'TO-DO' list constructed from previous documentation and user guides, provided grounds for reassessing the goals of the project this year. It was decided that the primary objective of this year's work would be to get the go-kart moving in a controllable fashion with the minimum amount of hardware repair and modification possible. A computer vision tracking algorithm developed separate to this project could then be interfaced to the go-kart to demonstrate its autonomous movement capabilities. This would then provide a working platform that could be improved upon in future years.

3.2 Hardware developments

3.2.1 Brake bracket

One of the main missing components was the brake actuator mounting bracket. This was designed by the team in 2011 but lost over the years. A new bracket was designed after consultation with the departmental lab technicians who had overseen the design of the original cart modifications. The bracket was manufactured by the department's workshop and was secured to the base of the cart in line with the brake pedal. The brake actuator was help by a pivot at one end of the bracket and constrained from moving in the horizontal direction. The actuator replaced the brake pedal, pushing on the cylinder, and was allowed to move in the vertical direction as it applied pressure.

3.2.2 Steering and limit switches

The steering limit switches were mounted on a bracket which could be connected to the base of the steering column. The bracket and switches had to be adjusted such that as the steering shaft rotated, the limit switches were triggered by the pointer at the end of the shaft in a safe and appropriate manner. This could be done by attaching and rotating the steering wheel rather than attaching the steering actuator and having to run the risk running the system uncalibrated. Once the steering actuator was in place, applying 24V or less directly to the steering actuator terminals from a current limited bench supply bypassing the motor driver was found to be easiest. Applying less than 24V merely caused a reduction in speed.

When the CANBus system was first powered-up on the go-kart it was powered via bench supplies. It was observed that the steering calibration process would not complete successfully without manually cycling actuator power half way through. A systematic approach was taken to isolate the problem, which was found to be that the setup documentation followed from previous years did not account for grounding of the motor driver boards back to the brake and steering boards. This simple problem caused calibration to be unsuccessful although it was repetitively successful when operating on bench.

While testing the previously written API through the GUI mentioned in section 3.3, it was observed that steering commands would not successfully execute when the go-kart motor was running. Approximately three seconds after the motor was started, no matter the speed, the steering board would stop responding. Again a systematic approach was taken to isolate the cause of the problem, which appeared to be motor interference causing the steering board be unresponsive. It was found that the entire go-kart frame was acting as an antennae and the steering bracket connection meant that only the steering board was affected. Securing the bracket with nylon bolts solved the problem.

3.2.3 PWM Student board

The PWM student board is an intermediate motor driver board that monitors the go-kart motor currents, brake position, and desired motor duty cycle and passes an appropriate duty cycle on to the H-bridge driver board. The PWM student board is what is used by students to interface with the go-kart motor control. When the API design and bench testing was done in 2012 the PWM student board control was not fully implemented or tested and the board was damaged in the process. This meant that the board had to be re-flashed this year.

The PWM student board runs an old ATmega8 processor and is flashable via SPI rather than USB or JTAG. Documentation from the previous year referred to using parallel port interfacing from a desktop lab machine however this method was no longer possible due to the lab machines being upgraded. A USB-SPI interface along with Atmel Studio was required to successfully reflash the chip. Using Atmel Studio also required updating the code to meet modern compiler standards as the previous code and functions used had become deprecated. Additionally, as the motor board communicates with the PWM student board over SPI, with the motor board acting as master, the slave select pin had to be permanently pulled low. This required a hardware modification in the form of a jumper to ground across the chip. Performing these tasks and modifications were made more difficult by the lack of correlation between schematics, PCB layout diagrams and the PWM student board PCB itself.

As the student board had not been previously tested, it was connected to a go-kart motor simulation station and the CANBus system on bench. The software on the PWM board was modified to limit the average motor current, peak motor current and maximum motor current over one second to department specified values. Due to a lack in calibration of the simulation device at the time of testing, high currents were never encountered. Similarly as the go-kart was effectively developed under no-load conditions (up on blocks), high currents not experienced at the low duty cycles used. When running on the ground, friction would add load to the motor hence requiring a higher duty cycle for the same speed. Therefore it is recommended that the PWM board current sensing functionality be tested again before running the go-kart on the ground at high speed.

3.2.4 Power supplies

During initial development and testing stages bench supplies were used to power both the CANBus boards and the actuators. The motor driver boards and proximity sensor also get power via the CANBus boards, which in total drew approximately 800mA. The actuators drew a peak current of approximately 1.5A combined under loaded conditions. It was decided to draw power to the CANBus boards and PWM student board in parallel via a 24V rail on the go-karts motor driver circuitry. This was a fused supply from the battery and enabled using the control switch on the dashboard. It was also advantageous as it meant the PWM board was powered-up at the same time as the CANBus boards, which avoided unsynchronised changes in PWM duty that were observed when the boards were powered at different times. The actuators were powered from a newly added independently fused line. The configuration was such that all control boards, actuators and the motor would lose power should the emergency stop button on the dashboard be pressed.

3.3 Software developments

3.3.1 Test GUI

The easiest was to test and demonstrate the functionality of the go-kart control system both independently and together was using a test Graphical User Interface (GUI) rather than running scripts through the command line. This was developed in parallel to each module of functionality on the go-kart and facilitated thorough testing of the system. It can also be used as a starting point or debugging aid during future development. As this was a GUI for testing and development purposes only, all emphasis was on functionality and considerations such as layout and colour were approached if absolutely necessary in the easiest, most time efficient manner. The GUI, pictured in Figure 2, allows the developer to set motor duty cycle, steering and brake to any allowable value by entering it in the required field. Standard functions such as 'brake on' and 'steer right' are included as push buttons for ease.

3.3.2 Speed Control

3.3.3 Computer vision interface

To further prove and demonstrate the functionality of the go-kart control system, it was interfaced to a checkerboard tracking computer vision algorithm [8]. The algorithm determined the pose of a checkerboard given its size, in three dimensions. The algorithm was written using OpenCV [9] libraries using C++ and operated on live video stream from the laptop camera. Using an external camera, perhaps mounted on the front of the go-kart, could be easily specified. The pose of the checkerboard was given in 3D translational and rotational vectors. The algorithm was modified slightly to print out the translational vector only.

An interface between the Python high level API and the C++ checkerboard tracking algorithm was then written also using Python. The Python interface file essentially forked itself, splitting into two separate execution threads; one 'child' to run the checkerboard tracking algorithm and one 'parent' to communicate with the go-kart CANBus control. A 'pipe' was created between the two threads, allowing data to be passed between them. The 'child' process then started the C++ tracking algorithm configured to pass its output, the translational vector, into the pipe. The parent process was configured to read the pipe and adjust the speed and steering such that the go-kart maintained a set distance from the checkerboard. As the checkerboard was moved away from the camera, the motor duty cycle was increased accordingly until a maximum duty was reached. As the checkerboard was moved closer to the camera the speed decreased accordingly and the brake was applied when the checkerboard reached a specified proximity. The PWM student board monitored the brake such that the motor was stopped when the brake was applied. Similarly, the steering responded to try and keep the checkerboard in the centre of the camera's field of view. Scaling was applied such that when the checkerboard moved to the edge of the image, the maximum turning angle was applied. However distance from the camera was not compensated for and this introduced an additional scaling effect.

4. RESULTS

The project was successful in achieving its goals of getting the go-kart moving and controllable via a laptop. Each module was individually controllable and via a test GUI which was decided to be the best way to demonstrate and test the functionality of the system while providing something beneficial for future developers. In addition, basic autonomous tracking of a checkerboard target was displayed. The Python API was successfully interfaced with the computer vision tracking algorithm and the motor, brake and steering all responded to changes in target position. The degree of response was however varied and not well controlled. Table 1 highlights which of the specifications laid out in the beginning of the project were successfully met.

Table 1: Specified Requirements

Fit boards and actuators onto go-kart	Steering adjustable within 1 degree	Can set or get speed to 1ms^{-1}
Calibrate brake actuator	Ability to return to centre steer anytime	Emergency stop button shuts cuts all power
Calibrate steering actuator	Motor and brake independently controllable	Controllable via laptop commands/script/algorithm
PWM board produces 2.5kHz appropriate signal	Sensor detects axle RPM	Complete calibration on power-up
Integrate python API with computer vision target tracking	Adjust steering based on target	Adjust motor and brake to maintain distance

Table 1 Guide	
	= Specification not met
	= Specification met

The speed sensor successfully detected the rotation of the axle and was able to convert this to RPM and ms^{-1} value. This value was maintained within 0.5 ms^{-1} using the proportional control system developed. However as this was only tested with the go-kart elevated on blocks, accuracy may change when running on the ground due to frictional effects and rough terrain. The only specification to really fail was the requirement to adjust the steering to within one degree. One degree adjustments could be made on the test GUI to which the steering would adjust. However a one degree change in steering wheel angle was unable to be confirmed. As the project progressed this was no longer seen to be very important in achieving the overall goal and therefore did not have priority.

5. DISCUSSION

5.1 Reflection on progress

From initial inspection of software, documentation and functionality it appeared that the API written in 2012 was complete, thorough. However during debugging stages of this project it was found that several parts were incomplete, false claims of functionality had been made and several undocumented 'hacks' had been implemented. For example there was no acknowledgement by the PWM student board of data transfers across SPI. Instead the data was sent twice using two different functions and assumed to have been received.

The reports from last year's work also suggested that the system was virtually ready to move onto the kart, with only calibration required. However after approximately one month of work, the majority of which involved becoming familiar with the various aspects of the system; the go-kart and various code modules, it became evident that goals would need to be revised. Getting the go-kart moving became the primary goal. This highlighted the large gap between bench testing and getting the system actually working with all components running simultaneously. Many problems such as interference, power-on timing, and grounding which were not experienced on bench and therefore not accounted for had to be solved. Due to the complexity of the system, the symptoms of these problems were often encountered when all parts of the system were running, making it difficult and time consuming to conduct even a systematic elimination process to identify the problem. This was the major time consuming factor in the gap between the system running on bench and go-kart.

The go-kart platform itself and its electronic motor with h-bridge control were not designed as part of this project but is one of a collection of go-karts belonging to the department that are used in power electronics courses. This introduced the need for close communication with the technicians who were in charge of the go-karts and knew their inner workings. This year all the go-karts are having their batteries and solenoids upgraded. Due to this process the go-kart previously modified for use with this project had no battery power supply for many months. Delays in this battery upgrade process (outside of the scope of this project) meant that it was not possible to get the entire CANBus system operating together on one go-kart. Bench supplies were used as much as possible however the motor and speed sensor had to be tested separately to the actuators on different go-karts as they required high current battery supply. This meant that issues such as interference from the motor, only detectable when all modules were running on the same go-kart, were not detected until very late in the development time period.

5.2 Future Work

The structure of this project with an overall goal towards developing an autonomous navigating go-kart means there is plenty of future work necessary. The work done this year provides a go-kart base controllable via a laptop through CANBus. This platform facilitates development in all areas of autonomous movement and navigation.

At a high level, the next step toward the overall goal would be to enable the go-kart to safely move about in an open space according commands from the user, a command script or computer vision target tracking as implemented this year. Safety of both the equipment and surroundings should be considered with robust procedures such as emergency stopping and battery isolation from a distance. Functionality could then be increased to allow navigation around objects and to and from specific locations. The go-kart could then even be used to fulfil a useful purpose such as parcel delivery.

Modern autonomous vehicles use a wide variety of navigational and environment sensing equipment and techniques. These include GPS, computer vision, infrared, ultra-sound, radar, lidar and inertial navigation systems [REFS]. Research could be conducted into finding a suitable combination of sensors that would allow the go-kart to effectively navigate obstacles and identify its surroundings. Development of GPS waypoint navigation, path-finding algorithms or computer vision based navigation could be conducted as an alternative, in parallel or as a completely separate project. A combination of multiple such sensory and navigation techniques is recommended to take full advantage of the small yet robust go-kart platform allowing movement through various environmental conditions.

A compete rework of the current PCB control hardware is also recommended. This could be done in parallel to other parts of the project focussing on navigation that would use the existing PCBs. The 2011 project team chose to use very small Molex connectors to connect peripherals and debugging equipment to the PCBs. These proved to be extremely inconvenient and caused connections to be easily broken. The need for new connectors along with the various repairs and work-arounds that have lead to general mess of electronics over the last three years calls for a complete rework of the PCB control hardware. Upon redesigning the PCBs the following considerations should be made:

- Allowance made for housing the boards in several casings around the go-kart with peripheral connectors placed conveniently around the board.
- Debugging and status LEDs should be included in addition to the numeric display.
- Possibility of wireless/Bluetooth modules for transfer of commands and/or data.
- Compatibility with the new PWM student boards currently being developed by the postgraduate students in the department – these do not use SPI communication.
- Choose peripheral connectors that facilitate easy connecting/disconnecting when debugging the system.
- Making all boards with the same PCB layout would be efficient and ease familiarity for debugging. Debugging and general purpose components should be attached to all boards, not just the components required by the primary function of each board, as done in 2011, as this causes inconvenience when debugging.

Additionally, several areas of the control system were found to be incomplete or lacking during the integration of the system onto the go-kart. Due to the aim of getting the go-kart movable and controllable whilst making the least modifications to previous work and hardware as possible, some of these areas were not fully fixed this year. For example, the USB connectivity to the boards is at times intermittent, however after several reconnection attempts connectivity is established and maintained. Additionally acknowledgement commands should be sent and checked for upon every data transfer.

CONCLUSION

The goal of this project was to integrate the previously designed CANBus control system with the go-kart platform and allow the go-kart to be controllable via a laptop. In addition, the go-kart was required to respond to positional data from a computer vision algorithm, following a target checkerboard while maintaining a set distance from it. The purpose of these goals was to provide a functioning development platform from which various further developments could be made in a range of areas. All of this has been achieved successfully to a satisfactory standard. However there is much room for improvement in the areas of software, hardware and overall system design.

This project most importantly highlighted the level of work and detail associated with moving past the bench testing phase of development. This especially applies in this case where each stage of development has been undertaken by different individuals, at different times, with different end goals. The need for clear, accurate documentation of new work as well as changes made to previous work was clearly evident. This project required a very holistic approach due to the many areas in which development had to be made.

REFERENCES

[1]Massachusetts Institute of Technology. (2009). Technology. Retrieved from: http://grandchallenge.mit.edu/technology.shtml

[2]H. Jenkins, W. Looman, S. Richards, Z. Taylor, “Team Ramrod / Mariokart Wiki”, https://github.com/team-ramrod/mariokart/wiki, accessed 1 April 2013.

[3]Looman, W. Jenkins, H., Richards,S., Taylor,Z. & Bainbridge-Smith, A. (2011). Software engineering practices in the mariokart system. Christchurch, New Zealand: Department of Electrical and Computer Engineering, University of Canterbury.

[4]Richards, S. Jenkins, H., Taylor,Z., Looman,W. & Bainbridge-Smith, A. (2011). Safety by design for the mariokart system. Christchurch, New Zealand: Department of Electrical and Computer Engineering, University of Canterbury.

[5]Taylor, Z. Jenkins, H., Richards,S., Looman,W. & Bainbridge-Smith, A. (2011). Development of a marker following system for use in the mariokart system. Christchurch, New Zealand: Department of Electrical and Computer Engineering, University of Canterbury.

[6]Wigley, M. (2012). Fly-by-wire go-kart engineering report. Christchurch, New Zealand: Department of Electrical and Computer Engineering, University of Canterbury.

[7]Weerekoon, N.A. (2013). Go-kart project Setup guide and operating manual.

[8]Weerekoon, N.A. (2013). Checkerboard pose and tracking: for use as simple fiducial marker and navigational aid. Christchurch, New Zealand: Department of Mechtronics Engineering, University of Canterbury.

[9]OpenCV. (2013). Computer vision library. Retrieved from: http://opencv.org/

[10]Weerekoon, N.A. (2013). Autonomous go-kart specification report. Christchurch, New Zealand: Department of Electrical and Computer Engineering, University of Canterbury.

[11]Wigley,M. (2013). University of Canterbury mariokart project for 2012. Retrieved from: https://github.com/mataamad/Mariokart2

[12]Jenkins, H., Richards,S., Taylor,Z., Looman,W. & Bainbridge-Smith, A. (2011). Embedded hardware design for an autonomous electric vehicle. Christchurch, New Zealand: Department of Electrical and Computer Engineering, University of Canterbury.