BAI

SENIOR FRESHMAN ENGINEERING

2E10 ENGINEERING DESIGN IV

BUGGY PROJECT

INTRODUCTION
At this stage of your Engineering studies you have been introduced to basic principles of electronics, electro-mechanical devices and computer programming. This Engineering Design Buggy Project will provide you with opportunities to investigate the practical application of these principles to solve a practical real-world challenge. To achieve a full solution to this challenge you must complete a number of connected software and hardware tasks. Thus, to succeed in this challenge you must work efficiently as a group.

BACKGROUND

Light rail systems are a common sight in cities across the world. In Dublin we have the Luas. The routes for these systems are defined by tracks laid in the ground. The system must coexist with other road users so the trams must be capable of starting and stopping in response to external signalling or operational circumstances. For instance, the trams must stop at stations and allow passengers embark and disembark.
The Luas must also stop at signal points, e.g. O'Connell Street, and wait until it receives a signal indicating it is safe to proceed. The system must also be able to accommodate breakdowns and outages and use alternative tracks/routes to most efficiently get a carriage to its destination.

PROJECT GOAL
The aim of this project is to create an autonomous vehicle that is under wireless supervisory control from a remote station (see Fig 1) and safely coexists with other trams. In effect, each group is required to design and implement a micro-simulation of the Luas light rail system.

OBJECTIVES
The project will introduce you to the challenge of electronic systems design & integration. The project is an example of '*hardware and software co-design'* and the scale of the task is such that it will require teamwork as a co-ordinated effort. The vehicle (from now on called the "Buggy") must meet a number of progressive design challenges:

Bronze Challenge: Single buggy capable of following main track twice in an anticlockwise direction under full supervisory control. Buggy must be capable of detecting an obstacle whilst following the track, coming to a halt if it does. The buggy must safely park in the parking bay. No external end-user manual control input is permitted once the initial start is signalled.

Silver Challenge: Two buggies on track following the "Rules" of the track. First buggy out follows main track twice anticlockwise and comes in to park. Second buggy out follows main track once anticlockwise and comes in to park. Buggy must park between lines in parking bay using ultrasonics. Either Buggy must be capable of detecting an obstacle and taking appropriate action when following the track. Both buggies under full supervisory control. No external end-user manual control input is permitted once the initial start is signalled.

Gold Challenge: The first part of the Gold challenge involves repeating the Silver Challenge. After both buggies park they must then synchronously rotate 180° and carry out the Silver Challenge pattern again, but in the opposite direction (i.e., clockwise around the track). The buggies must execute an undertake or an overtake during this clockwise traversal such that they ultimately finish up parked in reverse order. When going around the track in the clockwise direction, the restriction regarding having only one buggy in a section at a given time is lifted and students are encouraged to complete the clockwise laps as expeditiously as possible, but without collision. No external end-user manual control input is permitted once the initial start is signalled.

To meet these challenges you must focus on

i) the design and implementation of a Buggy and its associated line-following subsystems, and

ii) the communication with, and supervisory control of, the buggy and the rail infrastructure.

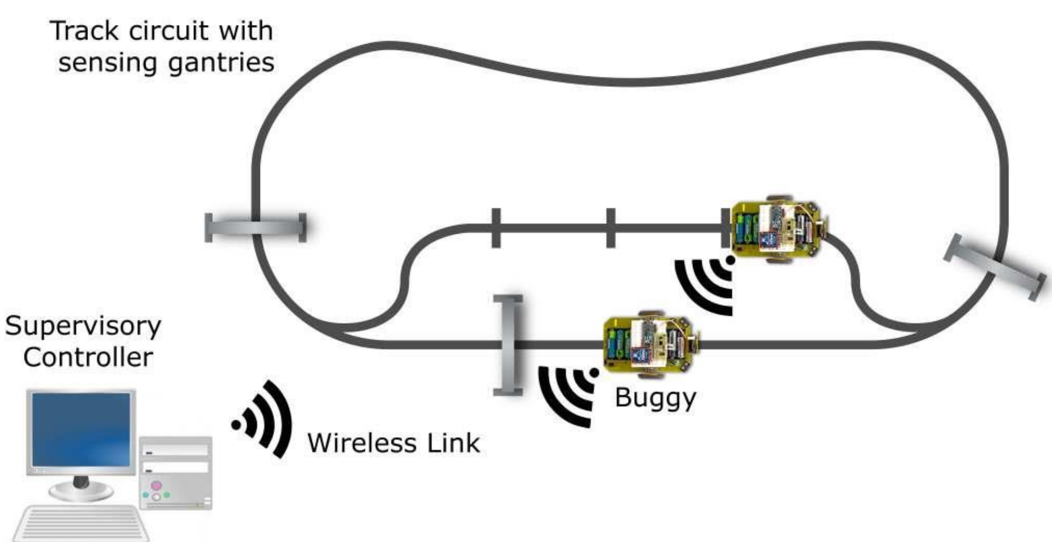


Fig 1. System Overview

Please note that in 2014/15 the track will have "turning arcs" in each of the parking bays to facilitate the 180degree rotation. These are not shown above, but are illustrated in the online animations on mymodule.

ORGANISATION

This is a <u>Group Project</u> and it is offered jointly by the School of Engineering (Electronic & Electrical Engineering) and the School of Computer Science & Statistics. The administrative task will be primarily managed using BlackBoard http://mymodule.tcd.ie/ . Here you will have access to support materials, task specifications, deadlines and report submission procedures.

The physical construction and system trials/testing work of the project will be carried out during weekly timetabled project sessions. Each Group will have two 2-hour supervised laboratory sessions and 2 x 1 hour lecture/tutorial sessions per week. All students are expected to spend approximately 12 to 15 ADDITIONAL hours per week working on the Project. This will include personal study to understand the technology and the challenges that must be addressed to deliver a successful outcome. The Reading Week, Week 7 of the Semester, will prove an excellent opportunity to refine the design and development work.

Each Group will have 4 or 5 members and must work as a group to ensure progress is made from Week 1. Groups are established by the Engineering faculty office and may not be changed or substituted (Group patterns E,F,G and H are in use). When the work is divided up, be sure to liaise and collaborate weekly as this will be required to fuse the development to arrive at the final working system. You must ensure all group members are fully apprised of, and understand, all elements of your solution. You will be assessed on this.

The early phase of the project will have two strands. These are

- Part S1 which will focus on the physical Buggy engineering and line tracking problem and
- Part S2 which will focus on the wireless communication and Buggy supervisory control engineering problem.

Both Parts will have to be fully integrated as the project progresses so the Group must remain closely engaged with the overall progress of their project goals and deliverables to ensure that both elements appropriately reflect their collective achievements.

Practical work on S1 will be primarily based in the Electronic and Electrical Engineering. The practical work on S2 will be primarily based in Computer Science and Statistics. The schedule and location of attendance for the Semester is provided in the Timetable. All students must attend at the specified sessions and ensure that they are checked in on the attendance sheet to record their presence. Groups will work as a unit towards completing the project and making the final submission.

However, individual tasks will be distributed among members of the group on a shared basis and each member in the group should understand all elements of the project. In particular, each group must meet regularly to co-ordinate the activities of Parts S1 and S2 and to ensure the project adheres to the agreed timeline.

ATTENDANCE and ENGAGEMENT

All students must attend all timetabled activities. Attendance will be taken and tracked on mymodule/blackboard. If you are ill or absent for medical reasons you must provide a medical cert from your doctor within 1 week of the absence. NO other absences will be excused.

Each uncertified absence, or advised failure to engage in the lab, will attract a 5% cumulative penalty for the individual on all subsequent group marks. For example, if you miss 2 sessions in the first three weeks then your mark (not the overall group mark) will be reduced by 10% for the week 3 deliverable and ALL subsequent group deliverables. If you then miss two more sessions between weeks 3 and 6 you will receive your group mark minus 20% thereafter, etc. If you are recorded as attending but failing to engage you will incur a 5% cumulative penalty for each instance.

Recognising that (on rare occasions) non-medical absences may be unavoidable, 5% credits may be earned by students who have already incurred penalties. To earn a 5% credit/offset against future penalty deductions, students must aggregate three consecutive penalty-free weeks of attendance and engagement. Students who successfully accrue three such continuous engagement weeks can ask for a 5% credit against their current percentage deduction total. This must be done in the week immediately following such eligibility being established. Only one credit will be granted for any such three week period. Credit periods may not overlap. e.g. a student carrying a 20% cumulative deduction from week 5 may, in week 9, seek a 5% credit for continuous engagement in weeks 6,7 and 8. If granted the student will then have an individual 15% deduction applied to all subsequent group marks from weeks 9-12. Neither credits nor deductions can be applied retrospectively.

PROJECT SPECIFICATION

You are required to conceive, design, implement and operate a system with remotely supervised autonomous Buggies. In Fig. 2 there is a representative description of the circuit and the basic rules under which the Buggies must operate. Note that these basic rules are superceded and disregarded as appropriate for the Gold task.

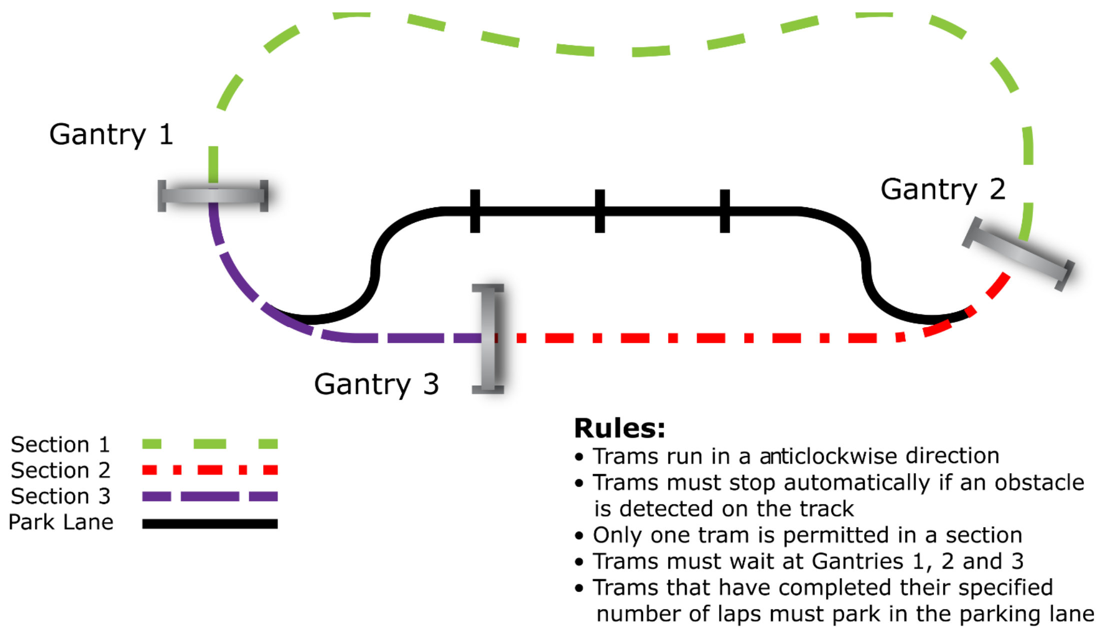


Fig. 2 Representative Description of the System

Note that the Rules above are amended for the Gold task as previously stated.

The project is an example of 'hardware and software co-design' and the scale of the task is such that it will require teamwork as a co-ordinated effort. You will have access to the basic shell of a Buggy that includes the motor assemblies, battery holders and sensors. The completed system, see Fig 3, should comprise of a Buggy Controller Unit which will interpret signals from the IR sensor pairs and include drive circuitry for 2 motor driven wheels. This means that the Buggy can be driven forward and steered by controlling the motors. The Buggy is provided with sets of infrared optical sensors to follow a line traced out on the floor. These sensor signals are to be connected as inputs to the microcontroller. The Buggy also has an infrared Gantry Sensor and Communications Unit which has an Arduino microcontroller and an XBee radio. This unit will communicate with the Base Station over a wireless link providing reports of the Buggy's location and receiving instructions to control its routing. It is also interfaced to the Buggy Controller Unit, through which individual motor control is affected. The Buggy also incorporates a front-facing Ultrasonic range measuring module. The Buggy can be configured to measure, and act on, the distance to an object in front of the Ultrasonic module.

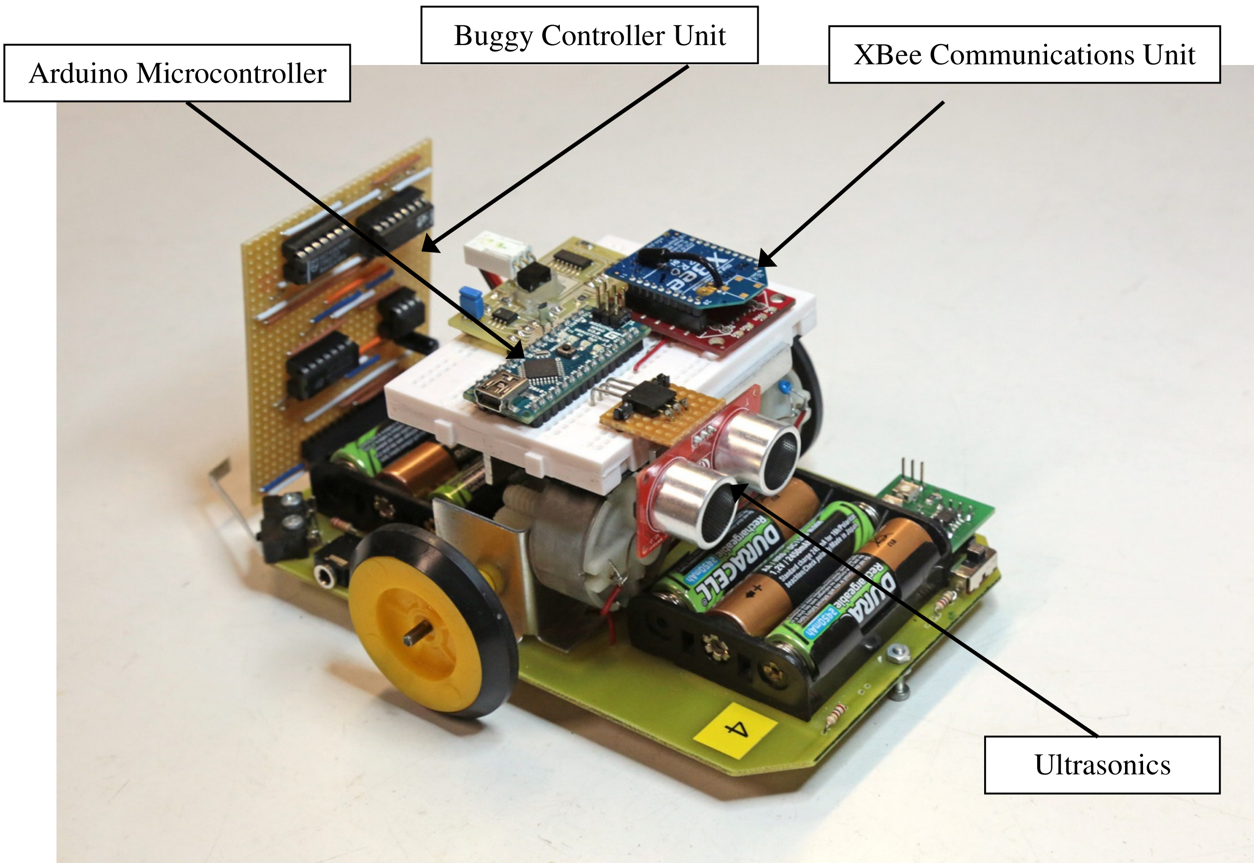


Fig 3. The Buggy and its subsystems

PROJECT ACTIONS

The challenges presented in the project specification simulate elements of the communications and control requirements of a light rail system such as the Luas. They are also relevant to the engineering challenges in the design and construction of autonomous vehicles used in many applications including warehousing, hospitals and assisted living. Each group will have access to a Buggy, a wireless communications module for the Buggy, a wireless communications module for a PC and collective access to a test track that includes signalling gantries.

The tasks to be performed comprise

- Buggy design and engineering;
- Communications and control systems and infrastructure.
- final integration, testing and evaluation;
- project planning including task identification and assignment of responsibilities, overall management,

Within each of these subtasks lie many individual tasks. Each group must identify and strategize for each of these tasks. The group will manage and co-ordinate the overall project activities and will be collectively responsible for the progress and success of the project.

S1: Buggy Controller
The S1 task will be responsible for the motor controller circuit design and construction. Part S1 will also be responsible for the design and development of the hardware to drive the Buggy. A small comparator circuit takes the sensor signals from the IR detectors. These outputs, along with override signals for the left and right motor are inputs to the circuit. A digital circuit must be designed in Multisim that uses these signals to steer the Buggy along the line. Each group will construct a circuit to implement their verified solution. Steps will include circuit layout and soldering. All team members will gain experience in circuit simulation and soldering.

S2: Communication, Supervisory
The S2 task will be responsible for work in 2 main areas:
Communications:
XBee modules are provided that can be used to easily establish wireless 2-way serial communications between the light-rail controller (PC) and the Buggies. The wireless modules will be used to transmit control signalling to the buggies and to receive signalling from the buggies. You will implement and test the bidirectional communications system between the controller and the Buggies. Thus you will have command control of one or more Buggies using a defined command vocabulary e.g. stopping, starting, branching. You will also handle gantry signalling received from the buggy as it passes each gantry.
Supervisory Control:
You will design, implement and validate the light-rail network controller. This task will be performed on a PC using a development environment of your choosing e.g. that which you are familiar with from 1E3/2E3. You will create the management and control module that is responsible for i) your Buggy and its actions; and ii) the whole light rail system.
In addition, your solution must integrate an ultrasonic unit that allows the buggy to detect obstacles directly in front of it.

Educational Goals:
We are providing a minimum problem domain specification for this element of the project. Therefore we are not telling you what to do. You will have to brainstorm as a complete group to identify the capabilities and functionalities each element of the project must have, and how they are will work together to deliver a final working buggy.
We provide you with a set of goals to achieve, and support and assistance at each step of the process. However we will not do the work for you.
Thus you will have to rapidly develop skills in areas such as small group working, realistic project planning, time management, goal delivery, human interaction and capability appraisal, etc. Refer back to 1E8. More particularly you should consider facets such as the perceived skills of each group member and how they map to the roles that the group have identified within the project. The group needs to produce a credible project plan, including deliverable task lists and due dates, and submit same by the end of week 3 for marking. An additional description of this task will be provided.
An interim project report should be submitted at the end of week 7 (which will also be graded). This should provide a critique of the original project plan, highlighting problems and successes, and providing an updated set of deliverables (with timelines) and a credible justification of the revised timelines in light of progress to date. A more detailed description of this task will be provided.

ASSESSMENT:

The Project will be assessed with both Continuous Assessment of interim deliverables and through a combination of demonstration, interview, short individual exam, peer assessment and Project Report. Details on the major assessment stages and reports will be provided separately.

Each group should retain a record of their work as it progresses. Code, etc **MUST** be uploaded to the CS SVN EVERY week. Manage your SVN repository carefully – this is the only record we will refer to for code validation. **Do not erase early versions of circuits or code. Take photographs/videos of your work as it progresses. Keep very regular backups. Keep snapshots of "working" code – particularly as the deadline and demonstrations draw near.**

IMPORTANT:

- Each group can access a take-home Arduino pack upon payment of a deposit. This will support the group's work outside timetabled hours in the Lab
- Each group must upload code to the SVN. This will be discussed further in CS. Code must be uploaded by a group at least weekly.
- The course is supported by the TCD VLE (Virtual Learning Environment) Blackboard. All course notes will be available through Blackboard. Your group should explore options for online discussions there and other ways of communication to improve team communication levels.
- You CANNOT repeat this module. Refer to the course handbook.
- Each group must have a laptop for the group demos in Week 12. All software should be loaded and tested well in advance.

APPENDIX A
Overview of Buggy

The Buggy consists of a chassis which is equipped with:

- 6 AA cells (1.2V rechargeable or 1.5V sandard) which provide unregulated power for ths IR sensors and the motors. The supply is regulated to 5V and supplied to the H-Bridge motor controller
- 2 bumper switches – unused
- 2 IR sensros for sensing and navigating along a line
- 2 motor/gearbox units each driving one wheel

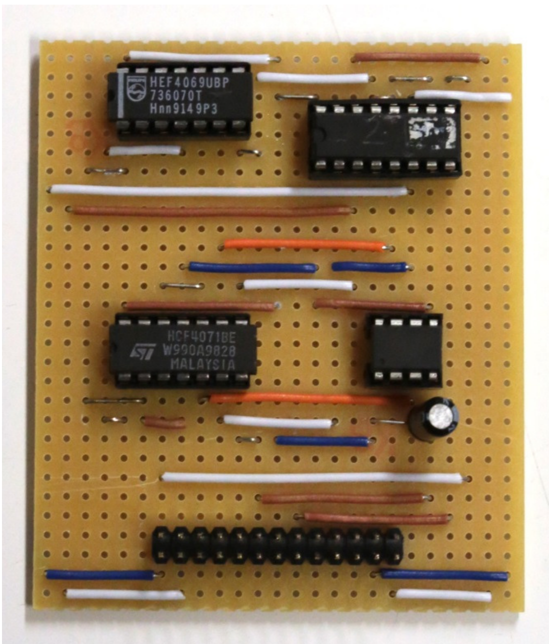


Fig A1 Sample Controller board with H-Bridge

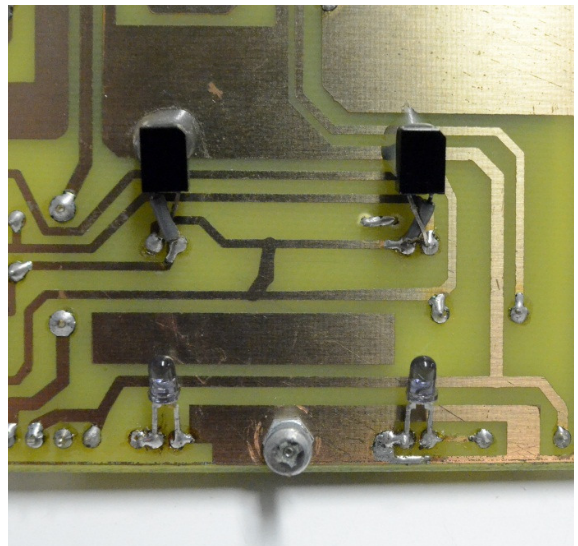


Fig A2 Underside of Buggy Chassis showing IR Tx and Rx

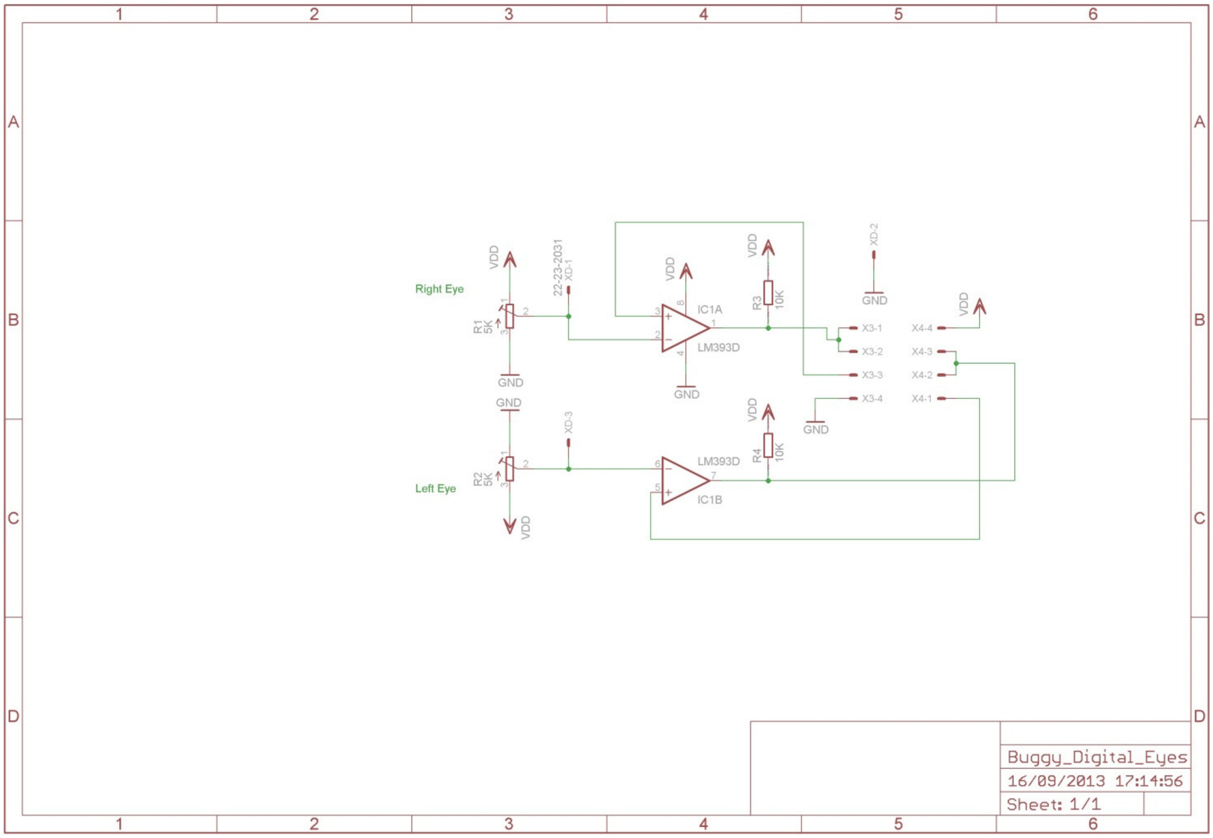


Fig A4 - Circuit Schematic for "digital eyes", taking IR sensor readings from Buggy chassis and converting to logic signals

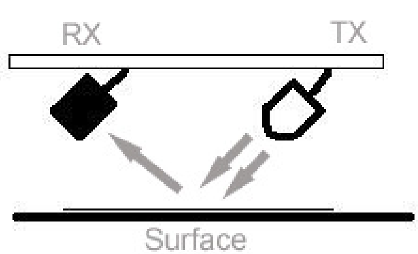


Fig A5 - IR Sensors functional diagram

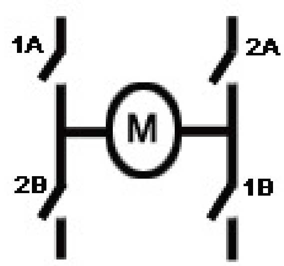


Fig A6 - H-Bridge functional diagram (Switches 1A & 1B act as a pair to drive the motor in one direction, 2A & 2B in the other) See

http://www.mcmanis.com/chuck/robotics/tutorial/h-bridge/index.html

There are two *IR sensors* located on the underside of the chassis. Each sensor consists of an IR transmitter and receiver. The principal of operation of the sensors is as follows:

- The transmitter (TX) emits a beam of infra-red light toward the surface along which the Buggy is travelling.

- The beam reflects off the surface and is then detected by the receiver (RX).
- The voltage seen is dependent on the amount of infra-red light detected by the RX meaning that if the surface is highly reflective, the voltage will be high whereas if the surface is dull, the voltage will be low.
- By sensing the variation in light intensity at the RX, the Buggy can determine if the surface under a TX/RX is black or white.

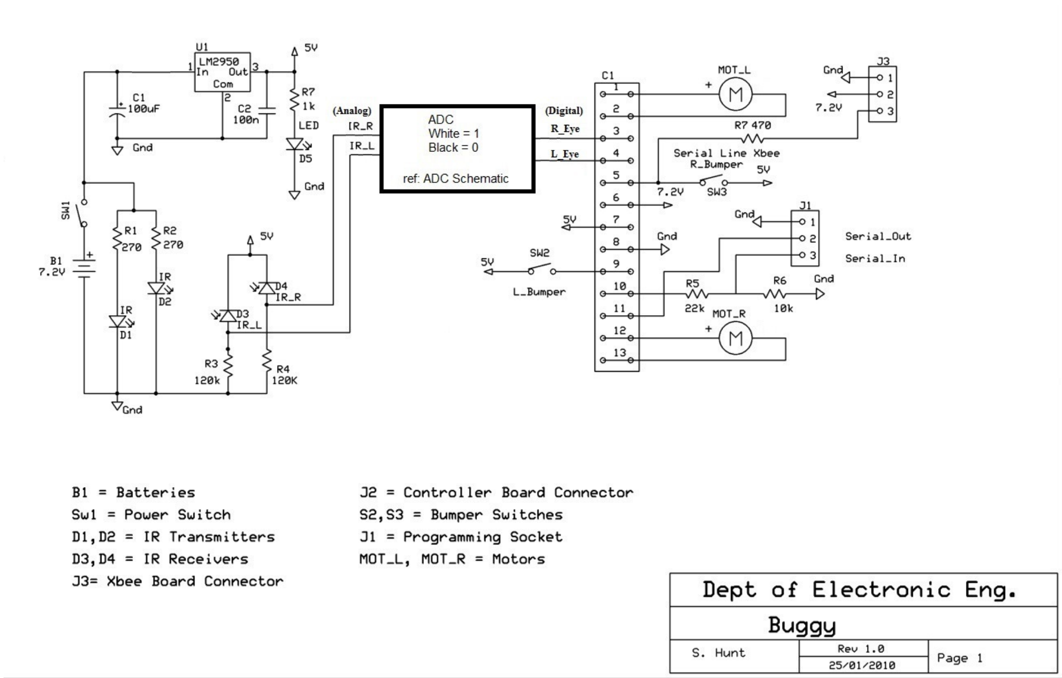
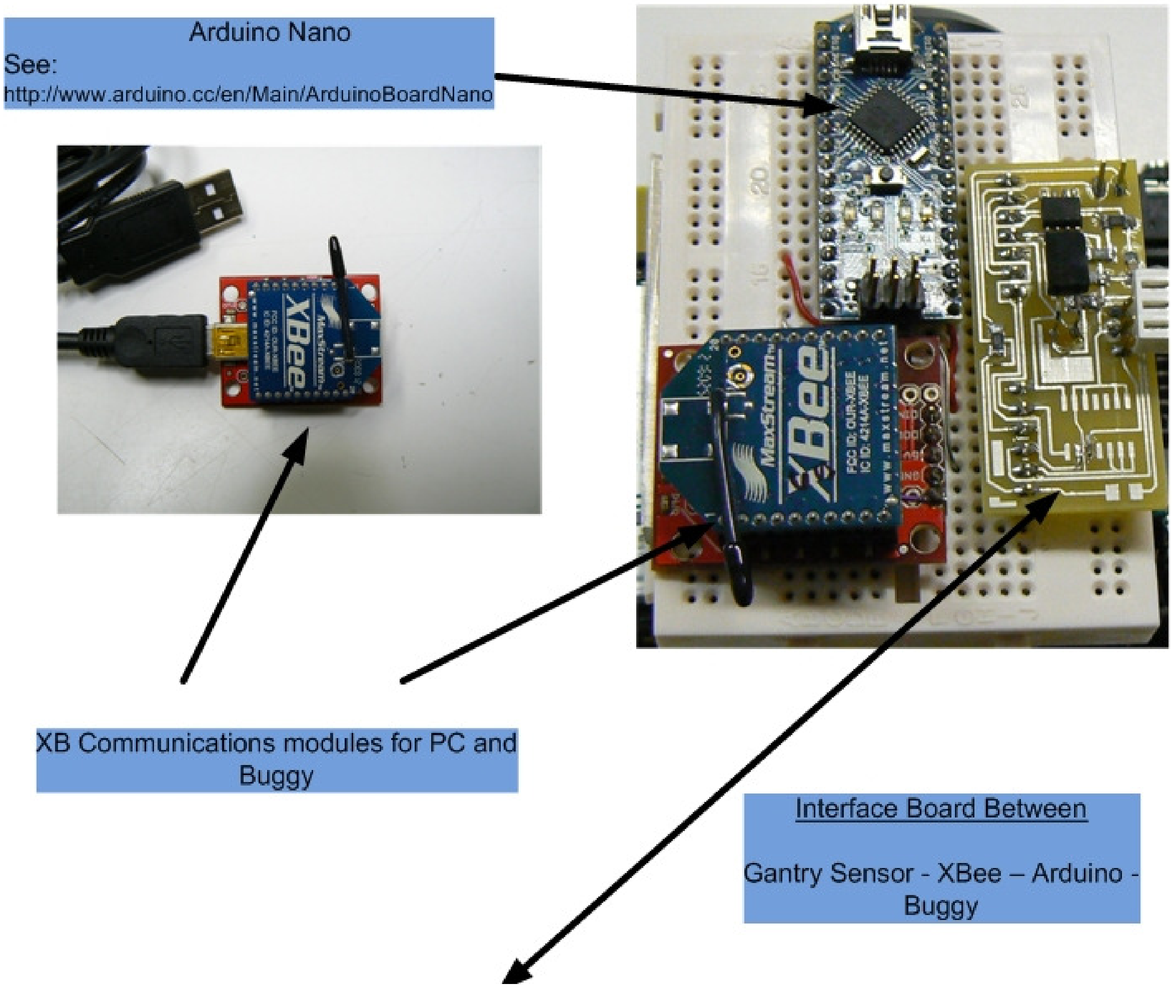
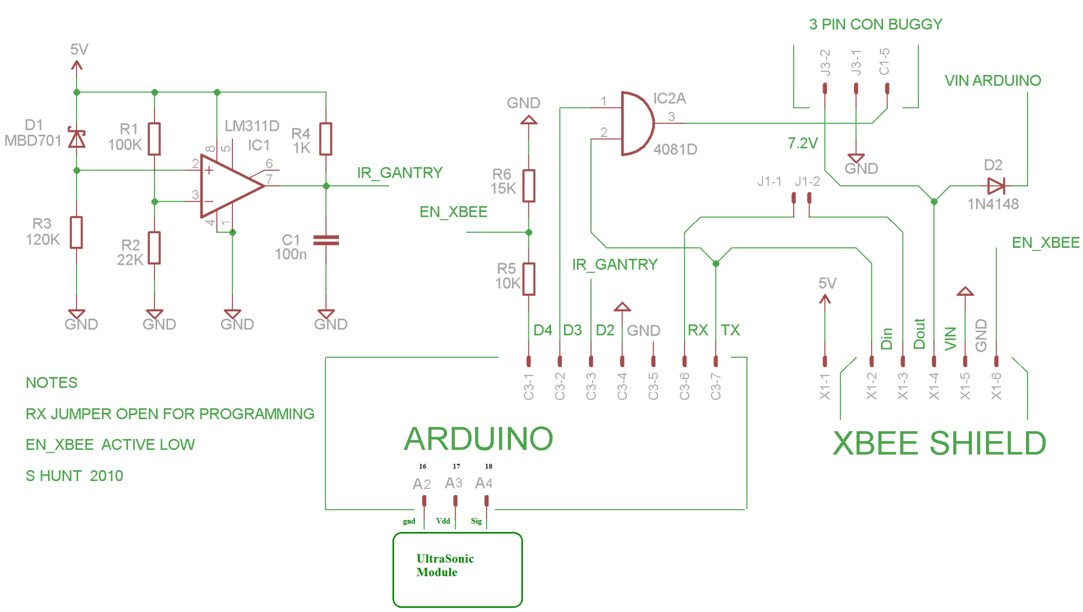


Fig A7 - Circuit Schematic for the Buggy chassis

APPENDIX B
OVERVIEW OF THE XBee COMMUNICATIONS MODULES





XBee modules

The XBee module is manufactured by the Digi Corporation (www.digi.com) and is an example of a point-to-multipoint wireless solution which allows device connectivity in a broad range of applications, including low-cost, short-range Personal Area Network (PAN) radios, high powered Frequency Hopping Spread Spectrum (FHSS) radios, and long-range Ethernet bridge radios.

Short-range wireless technologies are used for wireless communication in everything from remote monitoring installations (e.g. heat, security, light, shade, moisture, humidity, failure detection) to wireless telemetry (e.g. model aircraft, remote autonomous unmanned vehicles) to long range point-to-point wireless communication. Thus it is highly probable that you will all encounter situations and deployment scenarios in your future careers where wireless technologies can offer a viable solution set. Examples of short-range wireless technologies include Zigbee, WiFi and Bluetooth.

This lab will provide you with the opportunity to work with, and develop solutions using, wireless communications. The skills and experience you develop in doing so will be useful in your future years in College (regardless of the branch of Engineering you specialize in) and will also prove useful to your future employers.

The PC XBee unit appears as a serial port on the PC. To query and program the XBee unit you must identify the COM (Serial) port that the unit is connected to on your PC. Within your program you then open the COM port send your commands to the buggy unit. There are many code samples that will help you do this to be found on the Internet. Having established communications with, and initial control of, the buggy it is then necessary to understand how the XBee modules are configured. You will want to be able to communicate with your own buggy, and overhear others communications. This overhearing or eavesdropping, will allow your controller system to know what other buggies and devices are in different segments of the track. The demonstrators will provide you with more information on the AT Command Set for the XBee modules which is necessary for configuring the units to perform these tasks. Additional information is also available on the Digi website – www.digi.com

Development Environment

You can use any development environment that we can reasonably support within our labs. In the past most people have used the environment associated with their other CS and programming courses. Microsoft Visual Studio provides coverage for C++ and C# - two languages that have proved most popular in the past. In the first instance you should use a standard vocabulary of instructions that you will transmit from your PC station to the buggy via the XBee communication modules. 2 modules will be used. The module connected to your PC is the Sparkfun XBee Explorer USB module (www.sparkfun.com) The schematic diagram for this board is here: http://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Explorer.pdf and its shown in the image at the top left of Appendix B. The XBee modules can be purchased with either a"wire" antenna or a "patch" antenna. The main difference is in the transmit/receive range of the different packages. We use both and there is no functional difference for the purposes of this project.

The board contains a mini-USB socket which is connected to the PC via a USB cable. The PC USB socket supplies 5V to the module. A 5V FT232 USB ⇔Asynchronous serial signalling converter chip receives the data signals from the PC via the USB port and converts them to standard level-shift serial signalling. An onboard Voltage Regulator converts the 5V USB voltage line to 3.3V to power the XBee chip. Some LED's are connected to the signalling pins on the XBee to indicate when communication is taking place via the XBee.

The receiver XBee is shown alongside the Arduino Nano and the gantry detection board in the right hand image at the start of Appendix B. This red board it is mounted on is a very straightforward breakout board (shield) that has two connector pins to power the module as well as a transmit pin and a receive pin. This unit is inserted into a breadboard which is used to interconnect the module with the Arduino. The schematic for the breadboard interconnects, and the gantry detection board is provided in the bottom image at the start of Appendix B. We will ask you to examine this schematic and explain to us what functions the various circuits and components are performing.

Arduino Nano

The Arduino family of microcontrollers comprise a variety of general purpose devices that offer good capabilities at an affordable price. They are used for controlling everything from car alarms to medical devices to autonomous vehicles to wind turbines and solar cell orientations. Further information on the Arduino can be found at: http://www.arduino.cc.

Arduinos can be programmed using a variety of platform and technologies, and the default development environment is based on Processing. The programming language is Wiring, which has a C/C++/Java like syntax, and includes specific capabilities for working with low-level electronics devices. Essentially Wiring/Processing for the Arduino include all the libraries and functionalities you are likely to need "out of the box". Thus you will be able to focus on the algorithmic functionality you want your Arduino to realise, without overly worrying about having to provide lower-level functionality.

You will need to be careful to ensure that any code you acquire from external sources is clearly understood by all members of the group, and then develop it to satisfy your particular requirements. Please ensure you acknowledge or cite sources for any material you use.

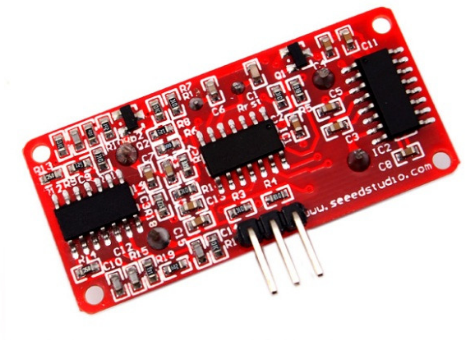
We will provide complete Arduino design and schematic information in a folder on Blackboard.

The Arduino needs to perform a number of functions. It must:

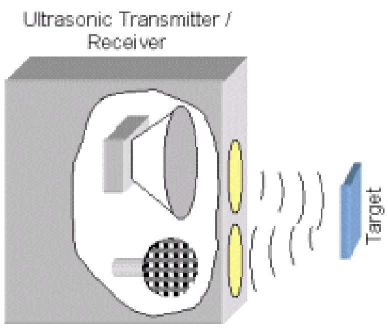
i) interact with the XBee module on-board the breadboard
ii) interact with the gantry detection circuitry
iii) provide signalling to the buggy controller unit
iv) interact with the ultrasonics module

The Gantries comprise infra-red transmitters that transmit infra-red signals of differing pulse widths. The Gantry detector board contains an IR detector and associated circuitry. Each Gantry produces a unique "signal". A signal from the Gantry detector board is delivered to the Arduino. The Arduino determines which gantry has been detected by measuring the width of the detected pulse. This information must be signalled back to the controller PC so it can be incorporated into the decision making strategy. The PC sends commands to the Buggy via the XBee and Arduino. The command vocabulary and corresponding motor declarations will be provided in a separate document.

APPENDIX C
Ultrasonic Module



.



(Source Euclid Research)

This is an ultrasonic sensor which can be used for distance measurement. This can be used to help park the Buggy and detect obstacles. Ultrasonic sensors work in a manner similar to radar or sonar. They evaluate attributes of a target by interpreting the echoes from radio or sound waves respectively. An ultrasonic sensor can generate a high frequency sound wave, transmit this out, and then evaluate the echo which is received back by the sensor. Thus it can be used to measure a distance to a target

For more information please view
http://www.seeedstudio.com/wiki/index.php?title=Ultra_Sonic_range_measurement_module