

A Secondary School Coding Project to Upgrade a Rampaging Chariot Sporting Robot with an Autonomous Sporting Capability & 'Mars Exploration' Capability Using a Raspberry Pi Computer and a Free Upgrade Kit

1. Introduction

Teaching programming skills to children is seen as a long-term solution to the “skills gap” between the number of technology jobs and the people qualified to fill them.

Pupils interested in STEM (Science, Technology, Engineering and Maths) find programming particularly engaging when it is associated with an exciting and practical real-time experience. The object of their programming should therefore physically exist, move and have a sporty performance.

A 'Rampaging Chariot' chassis provides an ideal basis for an autonomous vehicle with both sporting and 'Mars Exploration' capabilities. It is cost-effective, using an existing proven vehicle already available in over 200 secondary schools throughout the United Kingdom.



Standard Rampaging Chariot

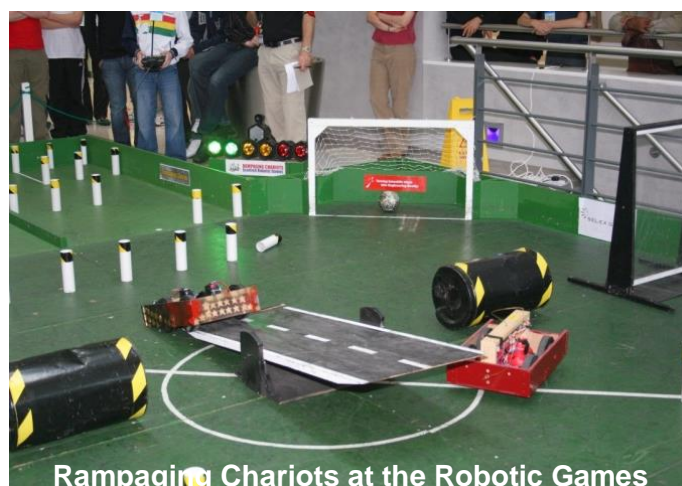
The basic radio controlled 'Rampaging Chariot' sporting robot is constructed by students from a comprehensive kit of parts. The new 'Autonomous Upgrade Kit' adds the capability of autonomous operation, and meets the coding educational requirements. It is a serious and challenging project for pupils who have successfully competed in the Rampaging Chariots Robotic Games.

1.1 The Basic Rampaging Chariot Sporting Robot STEM Project

Rampaging Chariots is an exciting robotic project sponsored by Leonardo and the IEE, aimed at interesting young people in the fun of technology and engineering by constructing radio controlled sporting robots from a comprehensive kit, to compete in a national competition.

It is a proven and popular STEM project that has been running for over 10 years and is well embedded in secondary school Young Engineers Clubs and youth organisations. It covers several diverse disciplines and is highly acclaimed by teachers. The project goal is to enter the annual Robotic Games hosted by Leonardo at their Edinburgh, Luton and Basildon sites, and to compete in four events: Assault Course, Sumo, Tug-of-War and Two-a -Side Football.

This four wheeled sporting robot is relatively large and powerful, and is powered by two 18V electric drills.



Rampaging Chariots at the Robotic Games

The construction task typically involves 4 pupils working for 12 one hour sessions, and includes construction of the chassis and assembly of two printed circuit boards. This is followed by testing, artistic enhancement and driving practice using the radio controller included in the kit.

1.2 The Autonomous Enhancement

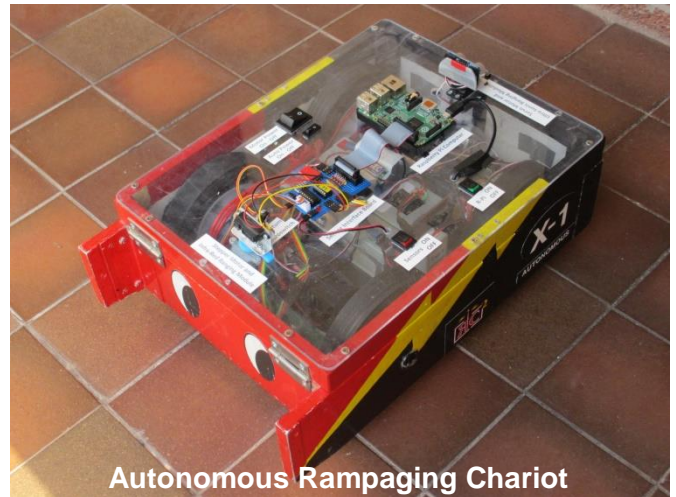
Participants in the Rampaging Chariots Project (both students and their teachers) have suggested that we extend this popular project to provide a versatile vehicle for secondary school pupils to learn coding with a practical and exciting outcome. As before, the project goal will be an annual competition, initially focussed on completing an assault course without human intervention.

The Rampaging Chariot chassis can be fitted with a flat 40cm x 30cm top which provides an excellent platform for a microprocessor such as a Raspberry Pi, together with fixed and scanning sensors of all descriptions.

The standard Rampaging Chariot chassis can be modified simply and easily for autonomous operation and provides an ideal platform for teaching practical computer programming skills and experiencing the result of real time experimentation with algorithms and sensors. Students learn that interfacing with the real world is a challenge, because real physical sensors do not provide perfect performance - unlike an icon on a computer screen.

The autonomous modification gives programming a practical real-time outcome and experience. The aim is to achieve an autonomous vehicle that will undertake tasks such as navigating around the assault course at the annual Rampaging Chariot Robotic Games. More advanced sporting challenges might include Sumo and Two-a-Side Football,

The project could also be extended to be a 'Mars Rover' with additional capabilities such as image recognition, a robot arm, and a soil sample analysis system.



Autonomous Rampaging Chariot



NASA 'Curiosity' Mars Rover

1.3 Autonomous Upgrade Kit Contents

- Raspberry Pi 3 Model B computer
- SD Card - Preloaded with the operating system and baseline autonomous software, written in Python and fully documented. This provides a starting point for personal modification and extension of the code. It includes supporting software development tools for Simulation, Visualisation, Telemetry and Datalogging.
- Sensors and Actuators
 - Two Odometers to sense the distance travelled by each drive wheel and allow robot heading to be determined
 - Infra-Red and Ultra Sonic Distance Measurers.
 - Stepper Motor and Servo motor for Scanning
- Interface Module - Protects the R-Pi from damage and facilitates connection to sensors of all descriptions.
- Power Converter (18V to 5V) and interconnecting Switches and Cables.

1.4 The Educational Aims

- a. **Vehicle.** A practical vehicle with a sporty performance to demonstrate and understand real time coding, control algorithms and mathematical concepts.
- b. **Computer.** A popular, versatile and powerful microprocessor with considerable on-line support resources.
(The two most popular microprocessors currently used by schools are the Raspberry Pi and the Arduino. We have chosen the Raspberry Pi as this is a complete computer and it has extensive publicity and web resources)
- c. **Software Language.** A computer software programme written in a popular language with full open access and on-line resources.
(Python has been chosen as it is generally considered to be the best language for pupils to learn. It is also the language most commonly used by undergraduates at universities.)
- d. **Code Modules** containing documented functions that allow the autonomous vehicle to be up and running quickly in a basic autonomous default mode.
These code modules provide the growth potential to stretch and challenge pupils as they use and modify the code and add extra and different sensors etc. to improve the accuracy, speed and capability. The modules are colour coded Red, Amber and Green to indicate the degree of difficulty and coding experience that is required to do modifications.
A software design that allows pupils to develop code off-line and visually see the simulated results on a computer monitor before practically testing the hardware.
- e. **Maximum Pupil Involvement** with the ability to share the hardware test vehicle by interchanging SD Cards. During development, students can store their work on a personal SD card to allow full traceability of modifications.
- f. **Teacher Support Resources** for hardware and software, including Video tutorials.
All abbreviations and concepts are expanded and explained in notes boxes or appendices.
- g. **Telemetry and Data Logging** capability to maximise the STEM teaching value by allowing pupils to visualise the effect of code modifications during development and testing.
- h. **Sensors and Actuators.** Different types of Sensors and Actuators to allow comparative performance to be demonstrated and assessed.
(We have chosen the following package:
 - *Magnetic Odometers to sense the distance travelled by each drive wheel and allow robot heading to be determined.*
 - *Infra-Red and Ultra-Sonic distance measurers*
 - *Stepper Motor and Servo Motor for Scanning**)*
- i. **Interface.** A simple interface to allow the sensors and actuators to be connected physically and electrically without difficulty. This should also provide the basic protection needed to prevent damaging voltages reaching the microprocessor.
Full circuit diagrams of each interface together with the relevant parts of technical data sheets.
Different serial data buses to demonstrate the features of each and allow pupils to interface other commercial sensors and actuators.
- j. **Dual Functionality.** The ability to switch the Rampaging Chariot between Standard Radio Control and Autonomous operation.

1.5 Software Coding.

We provide a sophisticated software programme, written in 'Python', that works 'straight out of the box'. Every effort has been made to simplify the code and provide extensive notes to explain how it works. For educational reasons we have simplified the code to reduce the performance of the example programme so that pupils can learn to code by experimenting in modifying the code provided to increase the speed, accuracy and sophistication of the code. This is to allow a real challenge to compete against other teams to create the fastest autonomous robot round the Assault Course at the Annual Rampaging Chariot Robotic Games.

Operating Modes

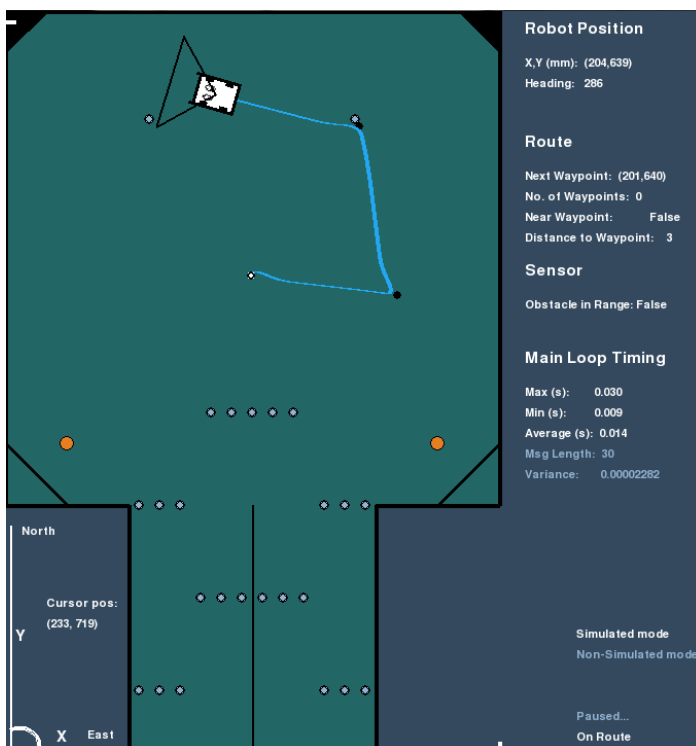
There are two available modes to use with the autonomous software:

- **Simulated Mode** is used for testing and debugging software code, without being connected to the chariot.
- **Real Mode** which requires the Raspberry Pi to be connected to the chariot as it will output to the motors and read from the odometers and other sensors.

The Graphical User Interface (GUI) is an excellent and powerful debugging tool that runs in both Simulated and Real Modes.

- It displays an accurate representation of where the chariot should be, in relation to the course.
- It shows diagnostic information such as loop timings and batch data sizes that can help you if the software doesn't run as expected.

- It also gives the users a means of interacting with the software during run time



The Assault Course Default GUI

The Autonomous Rampaging Chariot software is built using a robust design that functions through the use of multiple control loops. These loops are all run from a central control thread and each is responsible for a particular part of the robot and is run independently. The main achievement of the design is the ability to have a plug and play scenario where you can pick and choose “modules” that you can insert into the program. Each control loop is a part of the autonomous chariot system and allows pupils to make alterations that should not cause unforeseen changes to the Chariot performance in other parts of the programme.

The autonomous modification gives programming a practical real-time outcome and experience. The aim is to achieve an autonomous modification that will undertake tasks such as navigating around the assault course at the annual Rampaging Chariot Robotic Games. More advanced challenges might include Sumo and Two-a-Side Football, a robot arm, image recognition and a soil sample analysis system.

The modification sequence starts with the chassis lid and involves wiring, integration, testing and calibration of the sensors. The default system provides basic functionality, but there is huge scope for software coding modifications to improve the performance

1.6 Method of Navigation

Chariot navigation round an arena such as the assault course at the Rampaging Chariot Games is undertaken by navigating from waypoint to waypoint round the course. You designate these waypoints in x and y coordinates from the bottom left corner of the arena. The Autonomous Chariot travels from waypoint to waypoint in sequence by turns on the spot and straight lines. Curves/arcs can be added yourself at a later date to make the route faster and more efficient.

Raw Navigation Using Odometers

The distance travelled by the two drive wheels is measured by two odometers. Due to differences in motor power, wheel diameter and wheel friction, errors will build up between the required position in the arena and the actual position and heading of the chassis. Some of these differences can be measured and large errors corrected by applying calibration parameters, but we are operating in a real world, with a budget much less than NASA, and even they can never reduce the real world errors to zero (which would make the vehicle perform like a simulation on a computer screen).

It is these real world errors and uncertainties that make this project so interesting and provide us all with a real coding challenge. If it was simple it would be boring and you would not experience the satisfaction of beating this dumb Chariot with your human logic and innovation.

"Far better it is to dare mighty things, to win glorious triumphs, even though chequered by failure, than to take rank with those poor spirits who neither enjoy much nor suffer much, because they live in the grey twilight that knows not victory nor defeat."

Teddy Roosevelt

Test programmes are provided and are designed to simplify the programme and concentrate on the aim of testing and calibrating the chassis, motors, wheels and sensors to get the best possible performance. At the completion of the test programmes your calibration parameters will be inserted in the main programme and will therefore considerably affect the ultimate performance of your autonomous Rampaging Chariot.

Navigation Updates Using External Sensors

The raw, odometer derived position and heading of the Chariot is updated and corrected by using infra-red and ultra-sonic distance measurers. These sensors are mounted on motors that scan over a 180 degree arc to the front and back of the chassis. The sensors detect known obstacles and thereby determine the 'actual' position and heading of the Chariot in the arena. The 'actual' position is then used to update and correct the raw position.

Coordinate Scheme Used for the Autonomous Programme

The top of the arena is designated as North and our zero origin is in the South West corner. Position in the arena is measured in millimetres along the x and y axes. Angles are measured in standard compass format with zero degrees being North and angles increasing positive clockwise so that East is 90 degrees.

