King’s Legacy
RoboCup 2018
Team Description Paper

1.0 Team Details

Team Name	King’s Legacy
Organisation	Christ Church Grammar School, Western Australia
Country	Australia
Contact Person	Patrick Louden
Email	plouden@ccgs.wa.edu.au
GitHub Organisation	https://github.com/CCGSRobotics
Thingiverse Organisation	https://www.thingiverse.com/groups/ccgs-robotics

2.0 Introduction

King's Legacy, named after one of our founding members, is a student-driven group of robotics enthusiasts from Christ Church Grammar School in Western Australia. We began as a tight-knit group of passionate tinkerers but have since grown into a fully-fledged school society consisting of multiple teams focused on areas such as CAD/CAM production, electrical engineering and software development.

Our robots are based on the Open Academic Robot Kit and include some enhancements developed by our members including gamepad controls, improved chassis designs and custom electronics configurations. Areas of active development are the expansion of our robots' sensory capabilities, and the development of models diverging from the OARKit design. We have made all of our work open-source via GitHub and Thingiverse and is now being regularly updated to reflect our progress.

Last year, we competed at the Nagoya RoboCup, which highlighted the strengths and weaknesses of our system, and have since then made some changes towards working on the weaknesses.

3.0 System Description

3.1 Software Description

Our software follows a client-server model allowing communication and control of the robot using a gamepad connected to a laptop.

3.1.1 Client Side

On the client side, we have a main client file which:

1. Calls another file to set up the gamepad using the Python2 evdev module
2. Connects to the server by another file called connect.py using the socket module
3. Imports the functions to send data across the network from definitions.py
4. Sets the arm to its starting position
5. Imports another file called lib_threading1.py which starts a thread to allow two servos to move at once using the threading module
6. Loops through all events received by the gamepad
7. Based on the event codes, calls the respective methods from definitions
8. The functions from definitions send the relevant data to the server

3.1.2 Server Side

On the server side, the server file uses the SocketServer module and creates a class called MyTCPHandler which inherits from SocketServer. BaseRequestHandler, implementing a method to handle incoming data from the connected client and call functions from it. Functions from another file (emubot.py) are used to establish a serial connection to the servos and send commands to move them. The functions either call moveJoint, which moves a servo on the arm or moveWheel, which moves a wheel servo.

3.1.3 Sensory and Vision Code

Our main efforts in development this year is the implementation of sensors which allow the robot to gain information about its surroundings. These include:

- QR code reader
- Movement detector
- Hazmat sign detector
- Temperature sensor (reading of the surrounding temperature)
- Speaker (laptop sends audio to the pi, which plays that audio)
- Microphone (pi picks up and sends audio to the laptop)
- Carbon dioxide sensor

Some of these are currently still yet to be fully developed and implemented. However, we aim to have these finished before competition.

3.2 Human-Robot Interface

3.2.1 Camera Connection

We transmit the camera feed from the robot's Raspberry Pi camera using a bash script that is run through the terminal with Raspivid. The feed is then received by a laptop-side script and piped to the media player application for display.

3.2.2 Command Line Readings

As per our system from previous years, we have controlled the robot via a python program which gives constant updates as to the status of the robot. This year we are working on incorporating software which allows the driver to obtain readings from the various sensors, and software which allows the driver to activate the microphone and speaker.

3.3 Communications

The robot has a 5GHz Wi-Fi dongle connected to the Raspberry Pi to create a wireless access point. This allows us to establish a connection to the robot from a laptop and control it remotely while viewing a live feed from a camera mounted on the robot's arm.

3.4 Hardware Description

3.4.1 Raspberry Pi 3

The Raspberry Pi is our primary controller for all the parts of our robot. Last year, we upgraded from a Pi 2 to a Pi 3.

3.4.2 USB2AX Dongle

The USB2AX dongle receives instructions from the Raspberry Pi and converts them into instructions that control the servos.

3.4.3 Raspberry Pi Wide Angle Camera Module

The camera allows us to see what we are doing in the maze, without needing line of sight to the robot. We deliberately chose a fish eye lens because it gives us a better field of vision. The signal from the camera is sent via the Pi to the laptop from which we view a live image feed.

3.4.4 TP-Link TL-WDN3200 5Ghz Wi-Fi USB Dongle

We use the TP-Link dongle to create a wireless access point for communication with the robot.

3.4.5 7 Dynamixel AX-12A Servos

We use these servos to move the tracks, arm and camera on the robot. They are daisy chained to each other and controlled by the USB2AX

3.4.6 3A Turnigy UBEC Buzzer

This converts 12V power input from the battery to a 5V output that powers the Raspberry Pi and Open CM Board.

3.4.7 Push Button Circuit Breaker

This is used as a safeguard against power surges and short circuits. Power can be restored by pushing a button on the front of the chassis.

3.4.8 Servo Power Toggle Switch

We have installed a toggle switch that allows us to choose when to turn the servos on or off. This is useful when we are performing tests on the Raspberry Pi and don't want the robot to move around.

3.4.9 External Power Adapter Plug

There is an external power plug on the front of the robot, so we do not have to rely on a battery during testing.

3.4.10 MacBook Air Running Ubuntu 16.04

The laptop is used for developing and executing code that communicates with and allows us to control the robot.

3.4.11 Logitech Gamepad F-310 (Handheld Game Controller)

This is our primary means of issuing commands to the robot. It can be connected to the laptop or directly to the robot.

3.4.12 3x TMP36 Analog Temperature Sensor

The TMP36 is a low-powered wide range analogue temperature sensor that outputs a voltage proportional to the surrounding temperature.

3.4.13 Arduino Compatible Microphone Sound Sensor Module

This is a high-sensitivity analogue sound sensor with a digital output pin for when the sound intensity reaches its threshold.

3.4.14 MAX9814 Electret Microphone Amplifier with Auto Gain Control

The AGC in this amplifier means that loud sounds will be dampened and quieted, so they don't 'clip' the audio, while quiet sounds are loudened.

3.4.15 1x Mono Enclosed Speaker 3W 4 Ohm

This is a small enclosed low-powered speaker with decent audio volume and quality.

3.4.16 1x PAM8302 Adafruit Mono 2.5W Class D Audio Amplifier

This extra-small mono amplifier is surprisingly powerful with very high efficiency.

3.4.17 1x UART Infrared CO2 Sensor (0-50 000 ppm)

This component is based on NDIR (non-dispersive infrared) technology and automatically compensates for temperature.

3.5 Physical Design (CATIA)

We have a team dedicated to designing the chassis and moving parts of the robot. We develop component designs using the 3D design and engineering software, CATIA and fabricate them using a 3D Printer, Laser Cutter, and conventional workshop equipment.

After seeing the vast physical differences of the robots from the 2017 RoboCup in Nagoya, Japan, we (the physical design team) decided to build some prototypes of similar concepts. We are currently working on (and developing) three robots: an upgrade of our robot from the 2017 RoboCup, a robot that moves with six legs "hexapod" and a robot that manoeuvres with rotating flippers.

We aim to compete in the 2018 RoboCup with an upgraded version of the robot we used in 2017 but may use one of the other prototypes if they prove to function better.

We also aimed to put more sensors on the robot, involving the designing of mounts to support these sensors. These include a CO2 sensor, a temperature sensor, a functioning speaker, and other sensors to comprehend the robot's environment.

4.0 Application

4.1 Setup of System

Our robot and most of the components required to operate it can be contained within a single pelican case. Standard setup involves starting up the raspberry pi and remotely accessing it from the laptop to start the server application. The operator can then insert gamepad into the computer and run the client application from the laptop. This opens up a connection, allowing the driver to control the robot remotely with a gamepad.

4.2 Mission Strategy

Our main strength last year was our driver's reaction speed and our team's ability to respond to sudden technical challenges efficiently. This year, we aim to incorporate our sensors into the competition to supplement our driving skills.

4.3 Experiments and Testing

On the physical side, we are experimenting with two brand-new designs which diverge from the traditional OARKit robots: a flipper design and hexapod design.

The hexapod design is a six-legged walker robot, using 22 (3 per leg) servos and an RPi 3. It comes with a removable lid with easy access to the wiring underneath, such as the camera, RPi. There is a camera mounted to the front of the chassis to give visual aid to the driver. There will be a tail on the back that will provide a secondary camera, infrared sensor weight distribution assistance. It can be flipped over on the course with little to no driving effect.
The Flipper robot design has some adjustable wheel arms to help it traverse uneven terrain. It makes use of an RPi; its undercarriage comes off to provide access to the wiring harness. It is made out of acrylic. It has an arm on it with a mounted camera for maximum visibility.

Our sensing capabilities are currently in the development phase, with some completed and others on the path to completion. See 3.1.3 for a full list of sensor projects.

4.4 Team Composition and Specialisations

Our group primarily consists of two teams, the design team and the software development team.

Our designers are skilled in using CATIA, a 3D design and engineering software, and are experienced at recognising physical weak points and developing new strategies to minimise stress on components.

The software developers primarily work on ways to improve and optimise the capabilities of the robot and making it more intuitive to operate.

4.5 Software Packages

4.5.1 Development Environments

- IDLE (Python scripts)
- CATIA (design)

4.5.2 Python 2.7.1 Modules

- Evdev (gamepad control)
- Threading
- Socket
- SocketServer
- Serial

4.5.3 Python 3.5.2 Modules

- Numpy
- Imutils
- Cv2
- Zbar
- PIL

4.6 Open Source Contributions

4.6.1 Software

1. A python module that allows a driver to control the robot using a Logitech Gamepad, consisting of client and server sides. This software is available under the RoboCup-2017-Driving-Code repository on GitHub. The 2018 repository is still under development, with our code being updated and re-written.

2. Code to call upon sensors on the robot to give readings of information, send and receive audio and visually recognise relevant objects. This software is available under the Sensory-Abilities-2018 repository on GitHub.

4.6.2 Physical Design

1. A robot arm that contains housing for the battery
2. A flipped wheel chassis configuration, for reduced stress on the wheel servos
3. Two sets of custom rubber wheel tracks
4. A gravel guard we used at the 2017 RoboCup
5. The prototypes for the flippers and hexapod designs
6. A redesigned holder for the picamera on the arm

4.6.3 Electronic Components

Schematics for a servo master power switch with push-button circuit breaker. See Annex B.

5.0 Conclusion

Our original robot began as a reproduction of the OARKit robot. From our experiences at the previous two RoboCups in Leipzig and Nagoya, we have made significant improvements and modifications to the physical design and software functionality.

Between now and the competition, we plan to finish the development and implementation of our sensory capabilities into the design, and then focus on preparing the team to be able to drive and operate the robot before the competition.

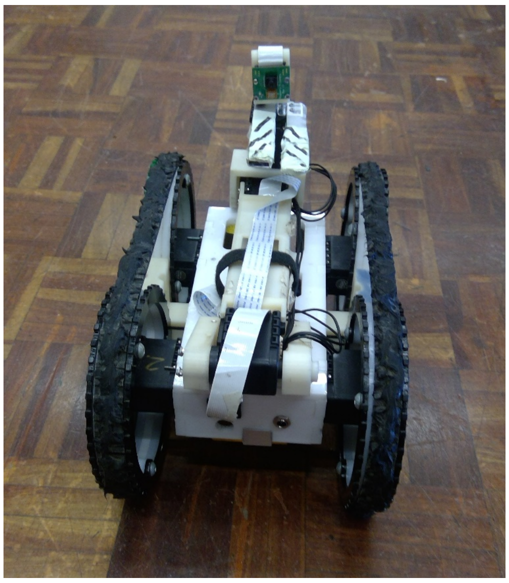
6.0 References

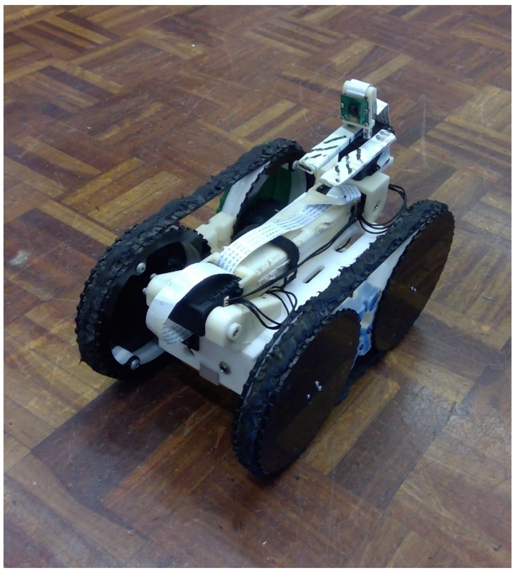
The Open Academic Robot Kit (OARKit)
http://oarkit.intelligentrobots.org

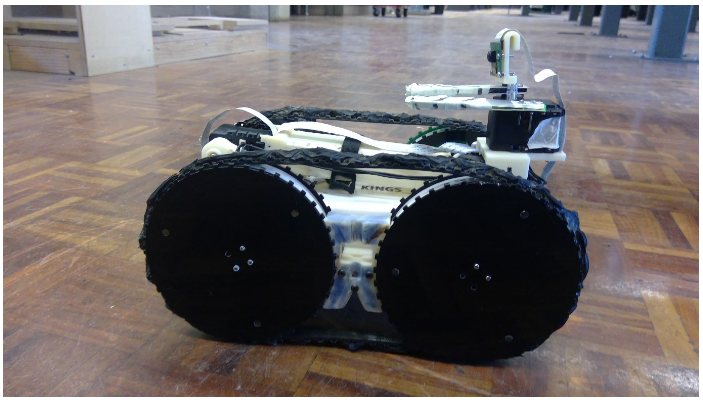
7.0 Appendix

Annex A – Robot Photos

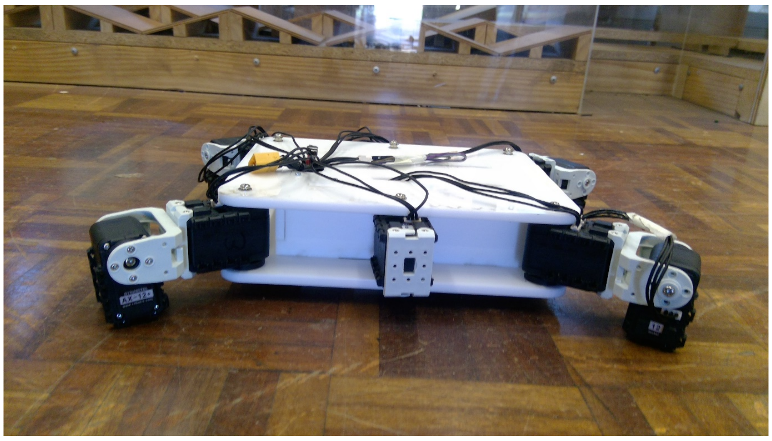
Main Robot (EmuBot)

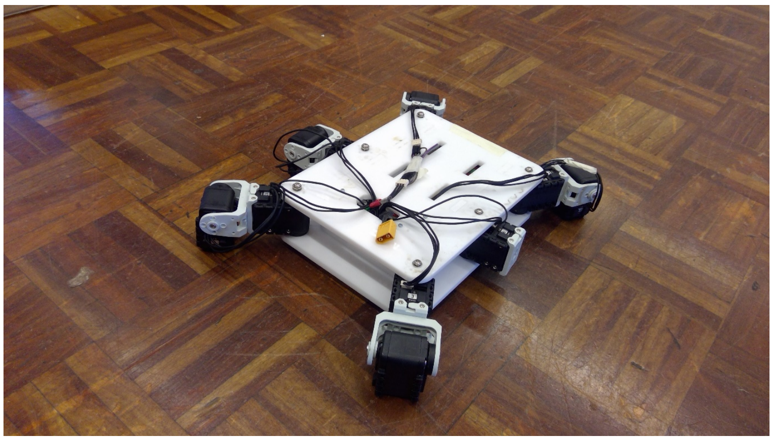


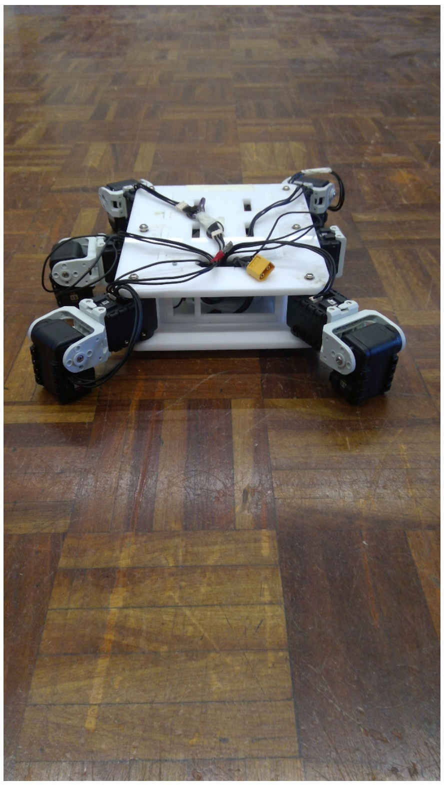




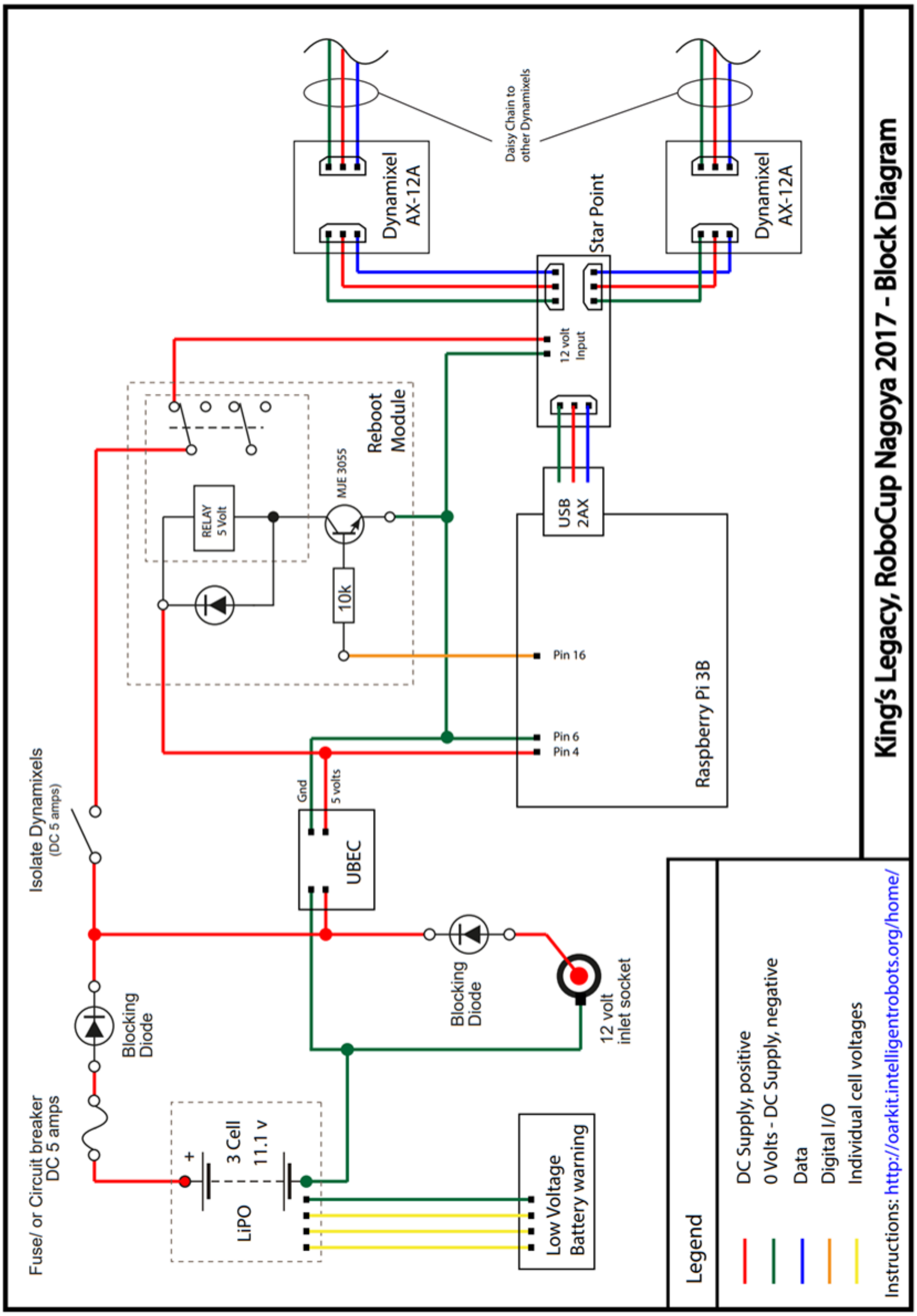
Hexapod Design Prototype



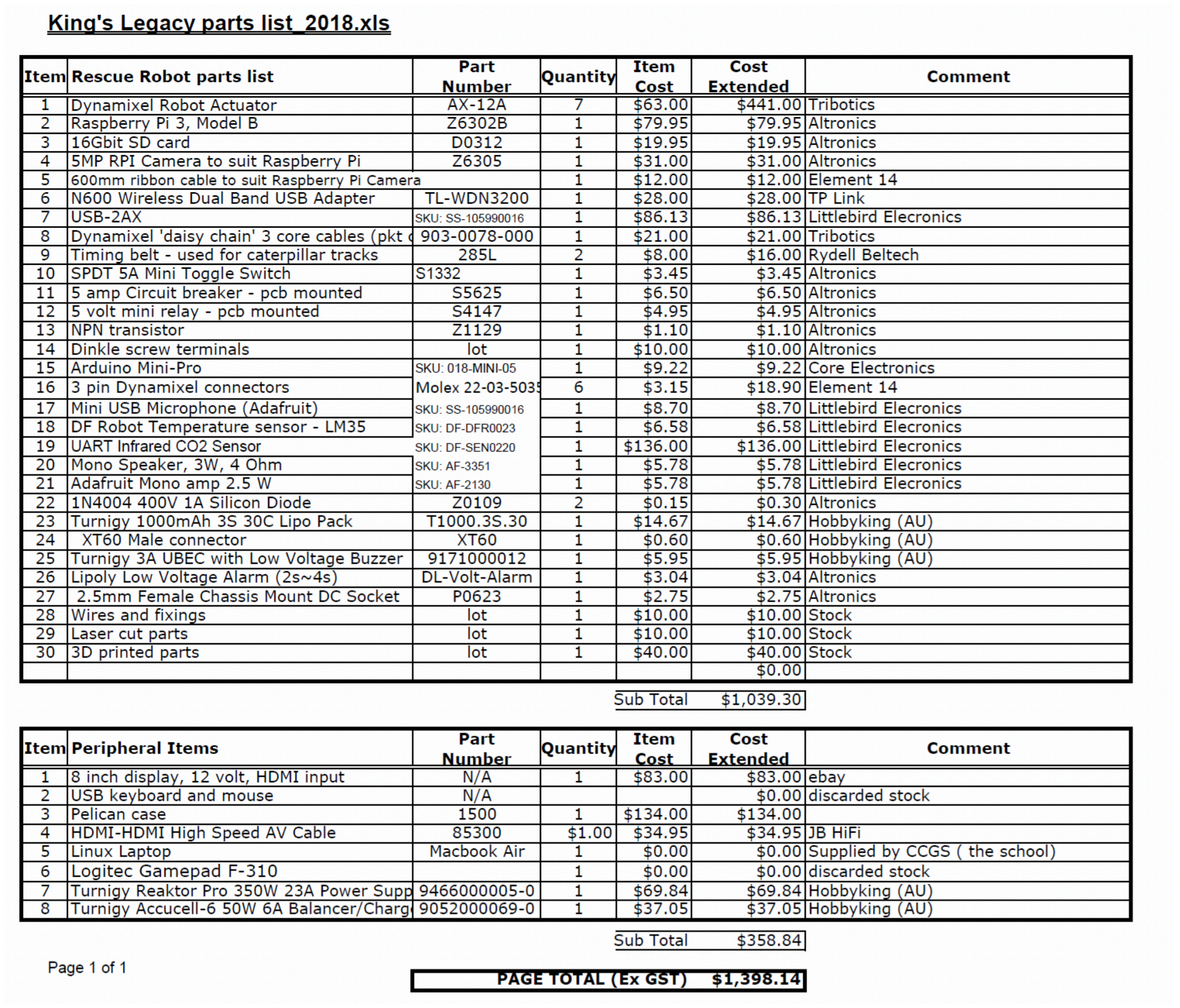




Annex B – Electronics Schematics



Annex C – Parts List and Costings



King's Legacy parts list_2018.xls

Item	Rescue Robot parts list	Part Number	Quantity	Item Cost	Cost Extended	Comment
1	Dynamixel Robot Actuator	AX-12A	7	$63.00	$441.00	Tribotics
2	Raspberry Pi 3, Model B	Z6302B	1	$79.95	$79.95	Altronics
3	16Gbit SD card	D0312	1	$19.95	$19.95	Altronics
4	5MP RPI Camera to suit Raspberry Pi	Z6305	1	$31.00	$31.00	Altronics
5	600mm ribbon cable to suit Raspberry Pi Camera		1	$12.00	$12.00	Element 14
6	N600 Wireless Dual Band USB Adapter	TL-WDN3200	1	$28.00	$28.00	TP Link
7	USB-2AX	SKU: SS-105990016	1	$86.13	$86.13	Littlebird Elecronics
8	Dynamixel 'daisy chain' 3 core cables (pkt c	903-0078-000	1	$21.00	$21.00	Tribotics
9	Timing belt - used for caterpillar tracks	285L	2	$8.00	$16.00	Rydell Beltech
10	SPDT 5A Mini Toggle Switch	S1332	1	$3.45	$3.45	Altronics
11	5 amp Circuit breaker - pcb mounted	S5625	1	$6.50	$6.50	Altronics
12	5 volt mini relay - pcb mounted	S4147	1	$4.95	$4.95	Altronics
13	NPN transistor	Z1129	1	$1.10	$1.10	Altronics
14	Dinkle screw terminals	lot	1	$10.00	$10.00	Altronics
15	Arduino Mini-Pro	SKU: 018-MINI-05	1	$9.22	$9.22	Core Electronics
16	3 pin Dynamixel connectors	Molex 22-03-503!	6	$3.15	$18.90	Element 14
17	Mini USB Microphone (Adafruit)	SKU: SS-105990016	1	$8.70	$8.70	Littlebird Elecronics
18	DF Robot Temperature sensor - LM35	SKU: DF-DFR0023	1	$6.58	$6.58	Littlebird Elecronics
19	UART Infrared CO2 Sensor	SKU: DF-SEN0220	1	$136.00	$136.00	Littlebird Elecronics
20	Mono Speaker, 3W, 4 Ohm	SKU: AF-3351	1	$5.78	$5.78	Littlebird Elecronics
21	Adafruit Mono amp 2.5 W	SKU: AF-2130	1	$5.78	$5.78	Littlebird Elecronics
22	1N4004 400V 1A Silicon Diode	Z0109	2	$0.15	$0.30	Altronics
23	Turnigy 1000mAh 3S 30C Lipo Pack	T1000.3S.30	1	$14.67	$14.67	Hobbyking (AU)
24	XT60 Male connector	XT60	1	$0.60	$0.60	Hobbyking (AU)
25	Turnigy 3A UBEC with Low Voltage Buzzer	9171000012	1	$5.95	$5.95	Hobbyking (AU)
26	Lipoly Low Voltage Alarm (2s~4s)	DL-Volt-Alarm	1	$3.04	$3.04	Altronics
27	2.5mm Female Chassis Mount DC Socket	P0623	1	$2.75	$2.75	Altronics
28	Wires and fixings	lot	1	$10.00	$10.00	Stock
29	Laser cut parts	lot	1	$10.00	$10.00	Stock
30	3D printed parts	lot	1	$40.00	$40.00	Stock
					$0.00	
				Sub Total	$1,039.30	

Item	Peripheral Items	Part Number	Quantity	Item Cost	Cost Extended	Comment
1	8 inch display, 12 volt, HDMI input	N/A	1	$83.00	$83.00	ebay
2	USB keyboard and mouse	N/A			$0.00	discarded stock
3	Pelican case	1500	1	$134.00	$134.00	
4	HDMI-HDMI High Speed AV Cable	85300	$1.00	$34.95	$34.95	JB HiFi
5	Linux Laptop	Macbook Air	1	$0.00	$0.00	Supplied by CCGS (the school)
6	Logitec Gamepad F-310		1	$0.00	$0.00	discarded stock
7	Turnigy Reaktor Pro 350W 23A Power Supp	9466000005-0	1	$69.84	$69.84	Hobbyking (AU)
8	Turnigy Accucell-6 50W 6A Balancer/Charg	9052000069-0	1	$37.05	$37.05	Hobbyking (AU)
				Sub Total	$358.84	

Page 1 of 1

PAGE TOTAL (Ex GST) $1,398.14