King’s Legacy
RoboCup 2019
Team Description Materials

1.0 Team Details

Team Name	King’s Legacy
Organisation	Christ Church Grammar School, Western Australia
Country	Australia
Primary Contact	Patrick Louden
Email	plouden@ccgs.wa.edu.au
Website	https://ccgsrobotics.github.io/
GitHub	https://github.com/CCGSRobotics
Thingiverse	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 Montreal 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

Our software system follows the client-server model in which the robot (Raspberry Pi) acts as the server and another computer acts as the client. This allows us to communicate with robot, monitor its sensors and control it from a client device (such as a laptop) with a connected gamepad. A significant amount of this year's development time was invested in enhancing this system.

3.1.1 Client Side

Our client-side application is a Node.js app built on the Electron framework. It performs the following functions.

- Connects to a Node.js server application running on the robot.
- Accepts a raw H264 streams and displays using JavaScript and a HTML5 canvas
- Uses a HTML5 controller API to get gamepad controller values.
- Sends plain-text instruction packets to the robot via UDP.

3.1.2 Server Side

The server application facilitates communication with the client using the **`SocketServer`** module. **`BaseRequestHandler`** allows it to handle incoming data from the connected client and call functions from it. Functions from another file (emubot.py) are used to establish a TTL serial connection to the robot's 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. It also performs the following functions.

- Executes system commands on behalf of the client via a Node.js interface layer. This negates the need for users to execute system commands on the robot via a separate SSH session as was previously the case. Common commands include:
  - Initiating the camera stream to the client.
  - Starting the Python server application which directly controls the robot's servos.
- Listens for instruction packets sent via UDP.
- Listens for web socket information via a compatibility layer script. This allows driving commands to be sent via http as well as from a native application.
- Automatically resets the state of the Pi when a client disconnects so that it is ready for the next client.

3.1.3 Sensors and Computer Vision

We continue to work towards improving our robot’s sensing capabilities.

Our current sensors are as follow:
- Temperature sensor (reading of the surrounding temperature)
- Carbon dioxide sensor
- Microphone (the Pi picks up and sends audio to the client)

The following capabilities are in active development:
- QR code reader
- Movement detector
- Hazmat sign reader
- Speaker (client sends audio to the Pi)

3.2 Camera Feed

We initiate a camera feed from the robot's Raspberry Pi camera with a bash script that calls ‘Raspivid’ and pipes the output to the client using ‘netcat’. As mentioned in 3.1.2, this script is executed by the server application when it receives a command to do so from the client. The client application then intercepts this stream and displays it in a HTML5 canvas.

3.3 Communications

We have written setup scripts that configure the internal WiFi adapter of the robot’s Raspberry Pi 3B+ to act as a 5GHz wireless access point. Client devices can then connect to this to establish communication with the robot. Our backup configuration relies on 5GHz USB WiFi dongle connected to the Raspberry Pi and configured to act as an access point.

3.4 Hardware Description

3.4.1 Raspberry Pi 3 B+

The Raspberry Pi is our primary controller for all the parts of our robot. Last year, we upgraded from a Pi 3 to Pi 3B+. The primary reason for this was to be able to use the internal 5GHz WiFi adapter as an access point instead of having to rely on an external USB dongle which consumed extra space and could potentially become dislodged on rough terrain.

3.4.2 USB2AX Dongle

The USB2AX dongle enables the server application on the Raspberry Pi to send instruction packets to the servos via a serial interface. This component replaces the OpenCM board used on the original EmuBot.

3.4.3 Raspberry Pi Camera Module

The camera allows us to see what we are doing in the maze, without needing line of sight to the robot. The signal from the camera is sent via the Pi to the client device from which we view a live image feed.

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

The dongle is used to create a wireless access point for communication with the robot. This has been superceded in our primary configuration by the interal WiFi adapter of the Raspberry Pi 3B+, however we continue to keep one on hand for emergency backup purposes.

3.4.5 Dynamixel AX-12A and MX-28 Servos

We use AX-12A servos to move the tracks, flippers and camera on the robot and a single MX-28 servo for moving the base of the arm. The servos are daisy chained to each other and controlled by serial commands sent from the Raspberry Pi via a USB2AX. We plan to swap the AX-12A servos controlling the flippers with MX-28 servos in the future for added strength and dexterity.

3.4.6 3A Turnigy UBEC Buzzer

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

3.4.7 Relay Controlled Circuit Breaker

This is used as a safeguard against power surges and short circuits. Power can be restored by sending a serial command from one of the output pins on the Raspberry Pi. This system is also useful for resetting the servos remotely if they become unresponsive on the track.

3.4.8 Servo Power Toggle Switch

This allows us to physically turn the servos on or off. It is useful when performing tests on the robot to prevent it from moving unexpectedly.

3.4.9 External Power Adapter Connector

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

3.4.10 MacBook Air Running Ubuntu 16.04

This is our primary client device which we use for monitoring and controlling the robot as well as for development purposes.

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 client device 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 Chassis Design

We have a CAD/CAM 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 at the 2017 and 2018 RoboCup competitions, the CAD/CAM team decided to test various prototypes based on our competitor's designs. After extensive testing we now favour a robot that manoeuvres using rotating 'flippers'. We have refer to this design as the 'Flipper Bot'.

The components of our current configuration can be summarised as follows:

- A flipped wheel chassis configuration, for reduced stress on the wheel servos.
- Flippers attached to each wheel for increased manoeuvrability and dexterity.
- Four sets of custom rubber wheel tracks.
- A gravel guard to prevent debris dislodging the tracks.
- A robot arm that contains housing for the battery.
- A grabber module mounted on the end of the arm for object manipulation.
- A mount for the Pi Camera built into the grabber.

3.5.1 'Flipper Bot' Chassis Configuration

Our Flipper Bot configuration debuted in the 2018 RoboCup Competition. Since then we have made several enhancements to the original design and aim to compete this year with an updated configuration that uses shorter flippers. Testing so far indicates this should give us greater manoeuvrability with less strain on the flipper servos.

The Flipper Bot's primary strength is its adjustable wheel arms that help it traverse uneven terrain. Other features of the design include a detachable undercarriage that provides easy access to the wiring harness and Raspberry Pi. It is made mostly of acrylic with a 3D printed arm mount similar to the original EmuBot configuration. The camera is mounted on the end of the arm for optimal visibility

3.5.2 Sensor Mounts

As we develop new sensing capabilities our CAD/CAM team designs solutions for mounting the necessary sensor modules. Current sensor mounts include a CO_2 sensor, a temperature sensor and a microphone. In development is a speaker mount and a laser pointer mount for the new grabber (see 3.5.3)

3.5.3 Grabber Module

Nearing completion is a new grabber module that we intend to mount on the end of the robot's arm. Once finished, this should significantly enhance our robot's object manipulation capabilities. The grabber mount will also include a mount for the camera and a laser pointer to assist in guiding the grabber to its target.

4.0 Operations and Procedures

4.1 Transportation and Setup in the Field

Our robot and most of the components required to operate it can be contained within a single pelican case. Standard setup involves switching the power on to the robot which activates the servos and boots the Raspberry Pi. A connection to the Raspberry Pi can be established by connecting to its self generated WiFi access point. Finally, connecting a controller to the client device and opening the client application enables an operator to monitor and control the robot remotely.

4.2 Team Composition and Specialisations

Our group primarily consists of two teams, the CAD/CAM team and the software development team.

Our CAD/CAM specialists 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, autonomy and ease of operation of the robot.

4.3 Development Tools

4.3.1 Software Tools

- Visual Studio Code
- IDLE (Python scripts)
- Atom Editor (GUI)
- Vim (GUI also, mostly for server-side operations using SSH)
- CATIA (CAD/CAM)

4.3.2 Python 3.5.2 Modules

- Numpy
- Imutils
- Cv2
- Zbar
- PIL
- Threading
- SocketServer
- Sys

4.6 Open Source Contributions

4.6.1 Software

Last year, our team used the Python3 code available under the repository at https://github.com/CCGSRobotics/RoboCup-2018-Driving-Code. However, with a push to move King's Legacy to a graphical interface, the main client/server interface has been changed from Python to a mixture of Python and NodeJS. The new repository can be found at https://github.com/CCGSRobotics/RoboHUD, where progress has been tracked using projects, issues and releases more extensively than in previous years. This repository is home to both the GUI client application source code and the server application code.

We maintain a separate repository for code associated with sensing capabilities. This can be found under the Sensory-Abilities-2018 repository on our GitHub page.

4.6.2 Physical Components

All of our physical component CAD/CAM files can be found on our Thingiverse page (https://www.thingiverse.com/groups/ccgs-robotics)

4.6.3 Electronic Components

Our wiring schematics which includes the relay controlled circuit breaker module are attached to this document (See Annex B).

5.0 Conclusion

Our original robot began as a reproduction of the OARKit EmuBot. From our experiences at the previous three RoboCups in Leipzig, Nagoya and Montreal we have made significant improvements and modifications to our robot's physical design and associated software systems.

Between now and this year's competition, we plan to further develop and implementation new sensory capabilities and focus on refining our procedures for navigating the robot on the terrain.

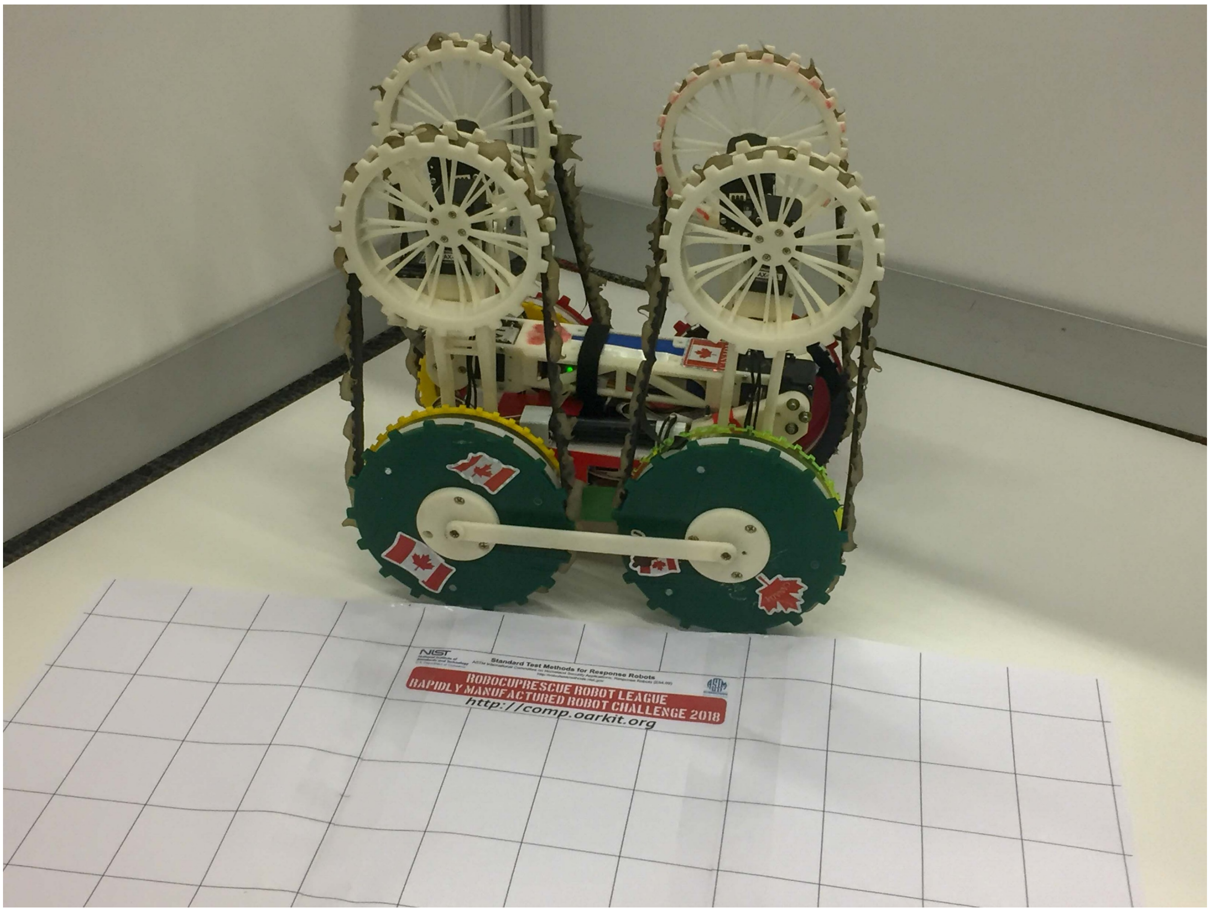
6.0 References

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

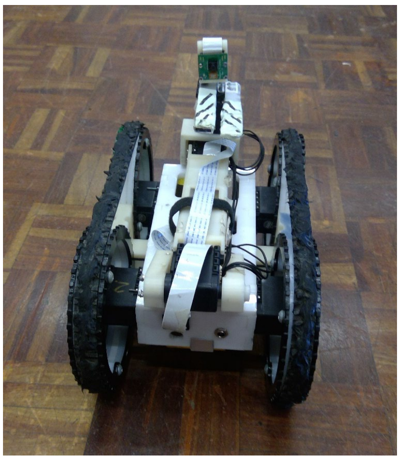
7.0 Appendix

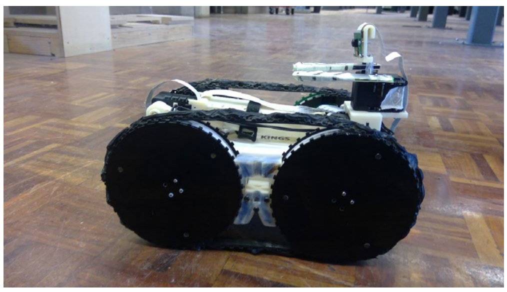
Annex A – Robot Photos

New Flipper Bot

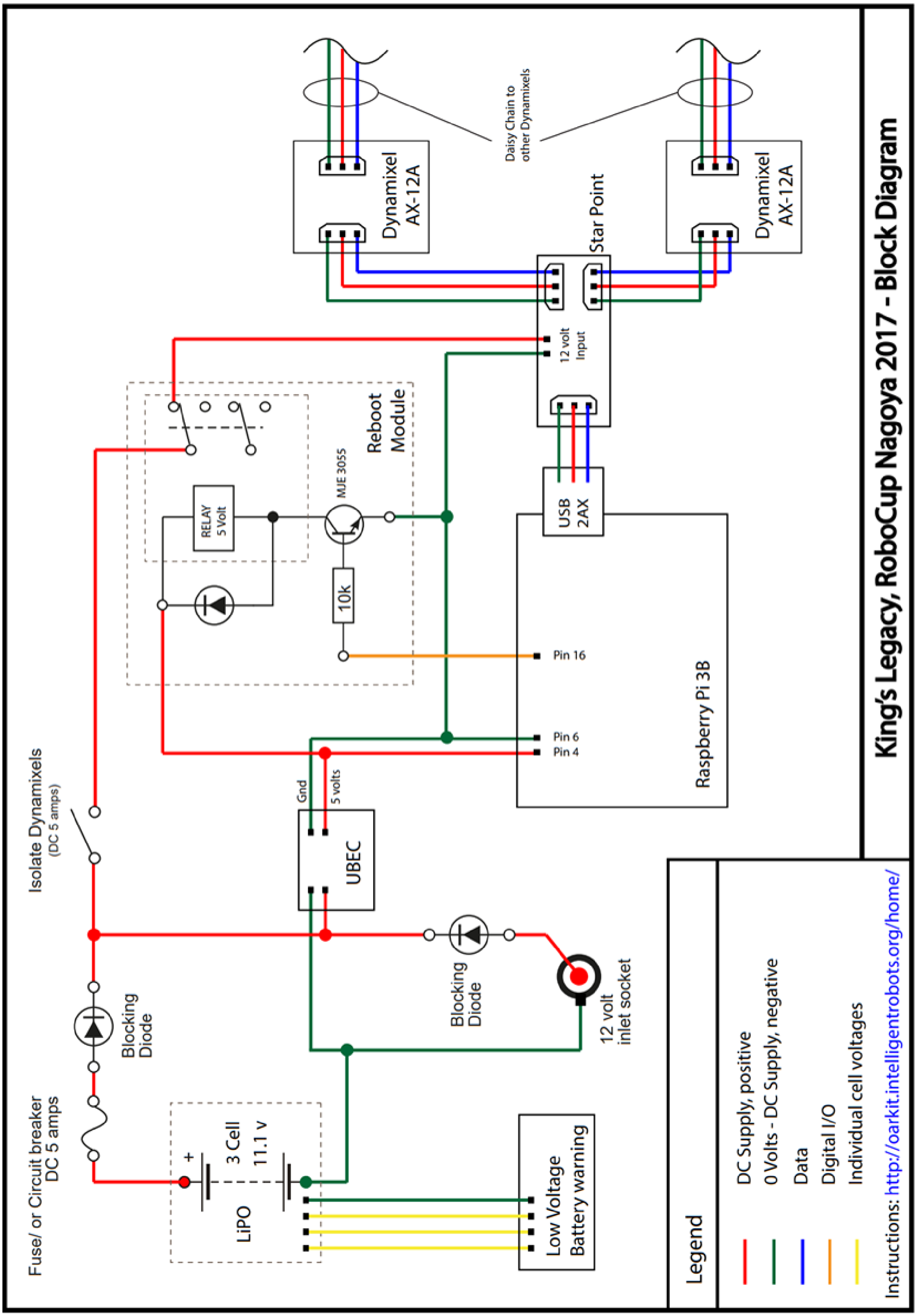


Original Emubot





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 22-03-5035	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 TOTAL (Ex GST) $1,398.14