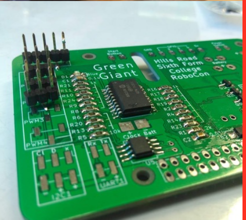
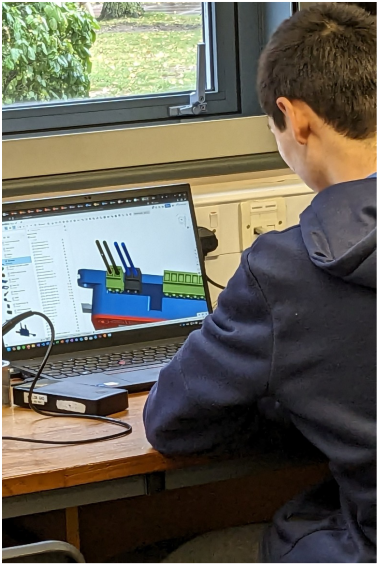


ROBOCON

2024

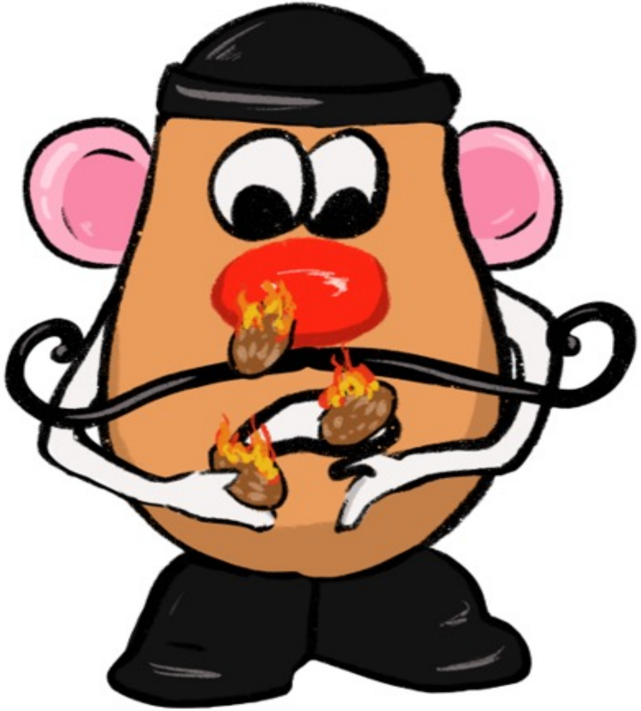
Robotics at Hills Road

Building RoboCon



ROBOCON 2024

Hot Potato



Spudsville – The Hot Potato Challenge

See page 4 of your rulebook for the full backstory…

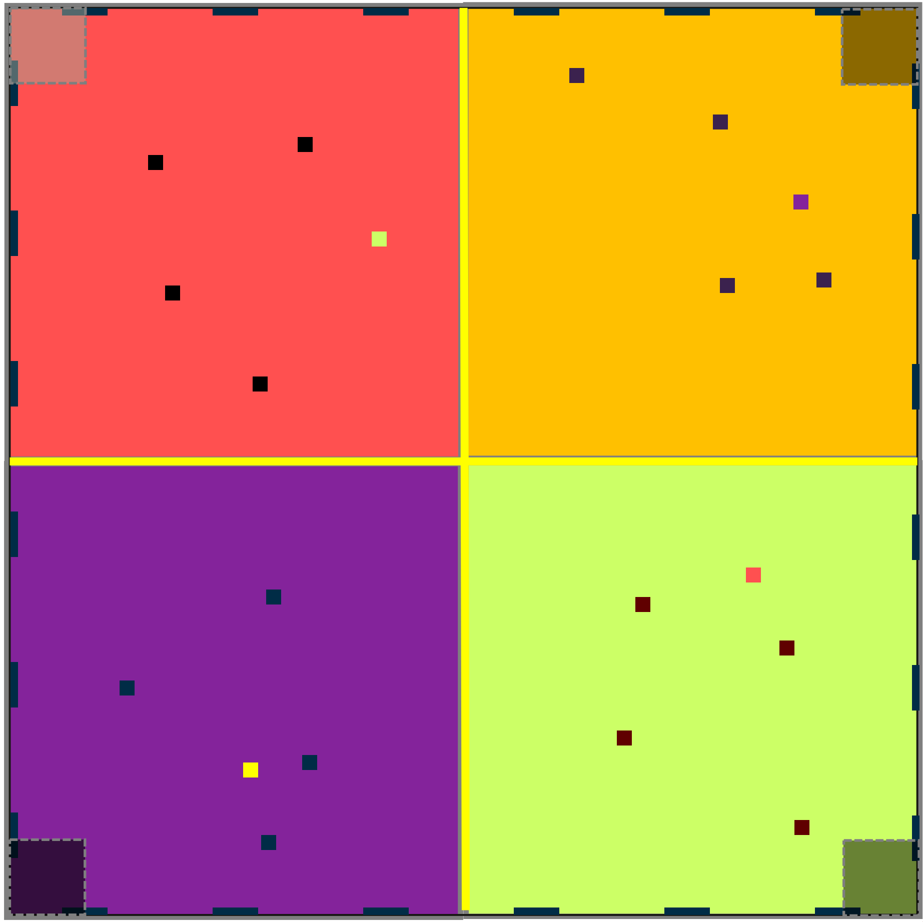
Hot Potato Rules

4 Robots

4 Jacket Potatoes

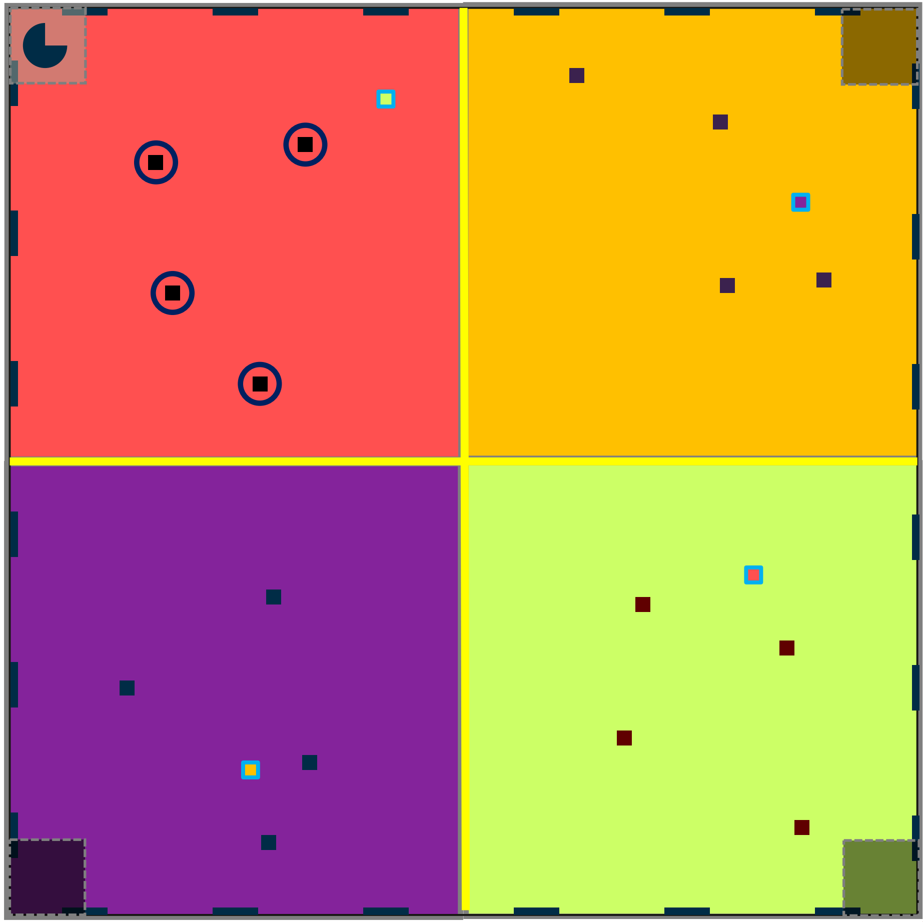
16 Hot Potatoes

3 Minutes to get rid of hot potatoes and/or collect your jacket potato



Earning Points

Potato Type	Location	Points Gained
Hot Potato	Your Patch	5 – [no. of hot potatoes] *
Hot Potato	Your Possession	0
Your Jacket Potato	Your Possession	2
Your Jacket Potato	Your Patch	4



* If more than 5 hot potatoes in your patch, 0 points are awarded

The Competition

Day 1	
Morning	Tinker Time
Afternoon	League Rounds

Day 2	
Morning	League Rounds
Afternoon	Knockout Rounds

The Competition

League Rounds

Competing robots will be awarded:

First:	8 points
Second:	6 points
Third:	4 points
Fourth:	2 points

Tied teams will receive the average of the position points

e.g. Two-way tie for 2nd – Both teams will receive 5 points

Knock out rounds

Robots will be seeded based on League points position

The two highest scoring robots will pass through to the next round

Winners of tied positions use League points to decide. Except for in the final at the discretion of Potato Head

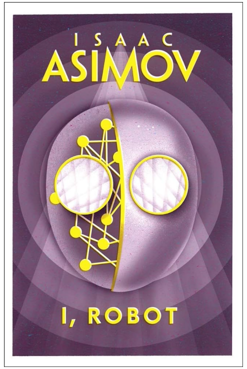
ROBOCON 2024

Game Q&A

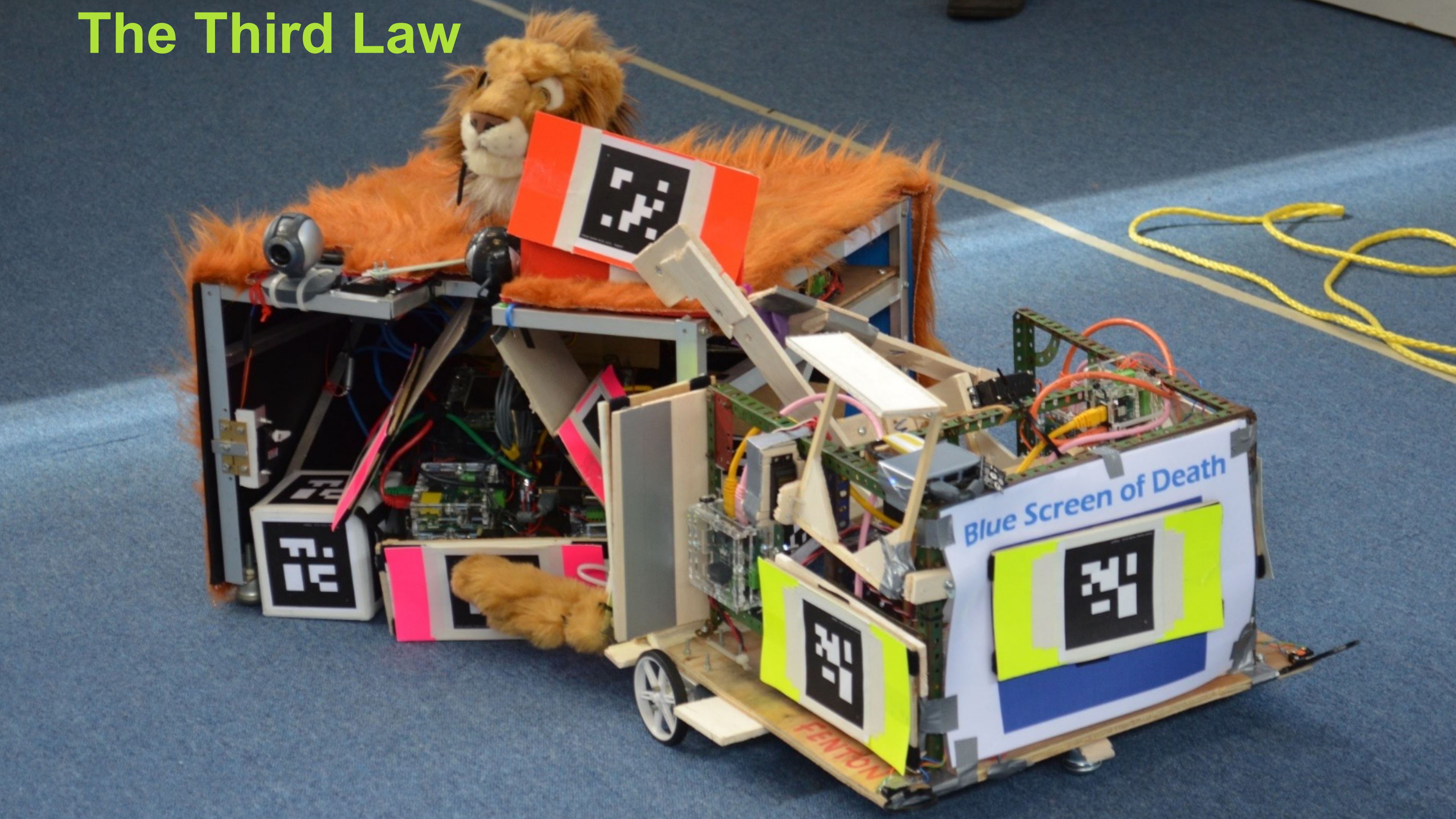
Asimov's Three Laws of Robotics

1. **A robot may not injure a human being** or, through inaction, allow a human being to come to harm.
2. **A robot must obey orders given it by human beings** except where such orders would conflict with the First Law.
3. **A robot must protect its own existence** as long as such protection does not conflict with the First or Second Law.

1. Robots should be designed so there is **no possibility of injury** to operators, marshals or the public.
2. Robots must not move until the start button is pressed, and **must include a prominent kill switch**.
3. Robots should be designed to compete in a non-combative manner, but should.



The Third Law



Awards

Main Competition Awards

Awarded to the top three teams in the final

(1st, 2nd and 3rd places)

Awards

Supreme Spuds

Given to the team that demonstrates that they have widened the field of possibilities through either an ingenious technical or logical development

Awards

A-Peeling artistry

Given to the team with the most robot and team flair

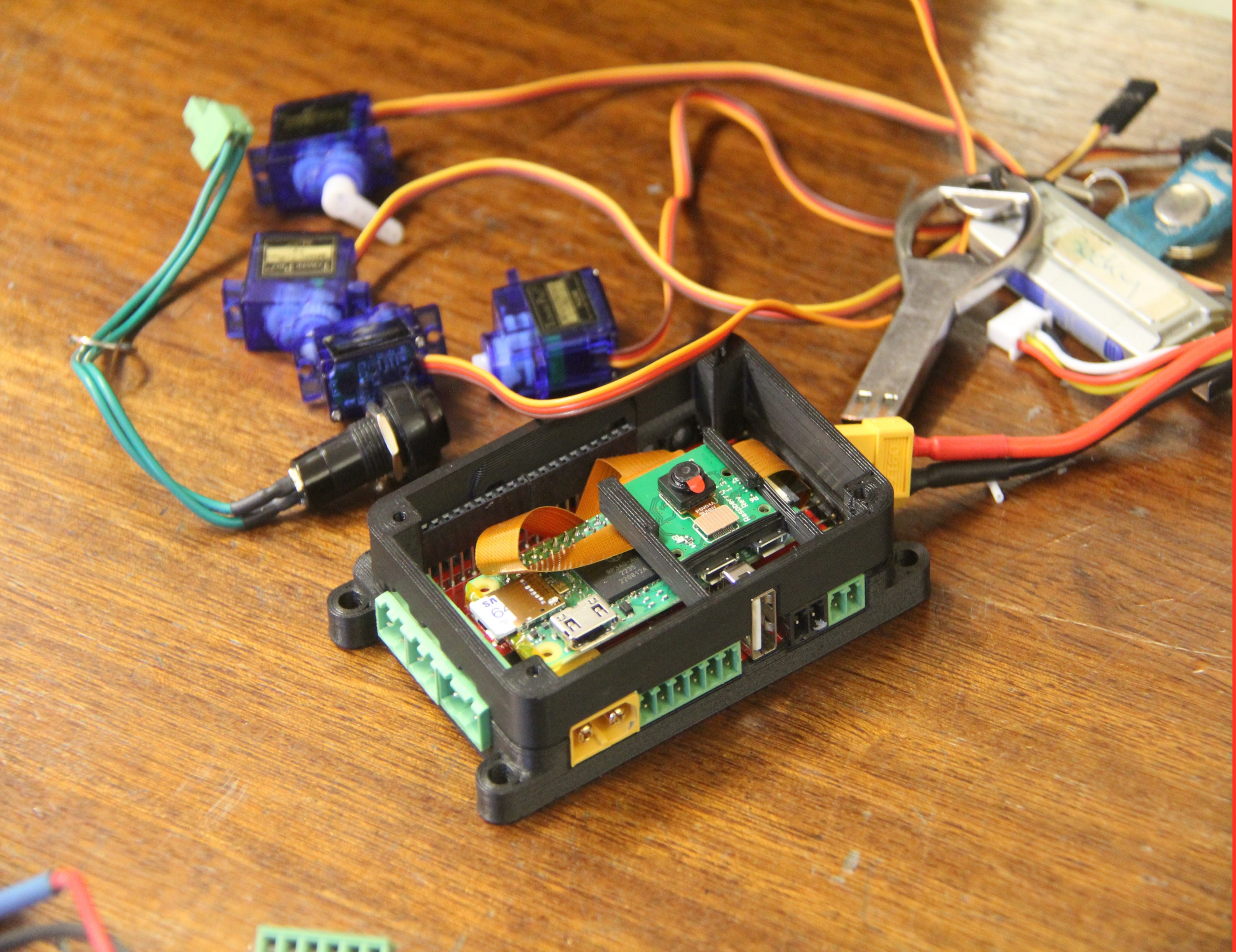
Awards

When the chips are down, you work your hardest to ketchup

Given to the team who shows perseverance in the face of adversity

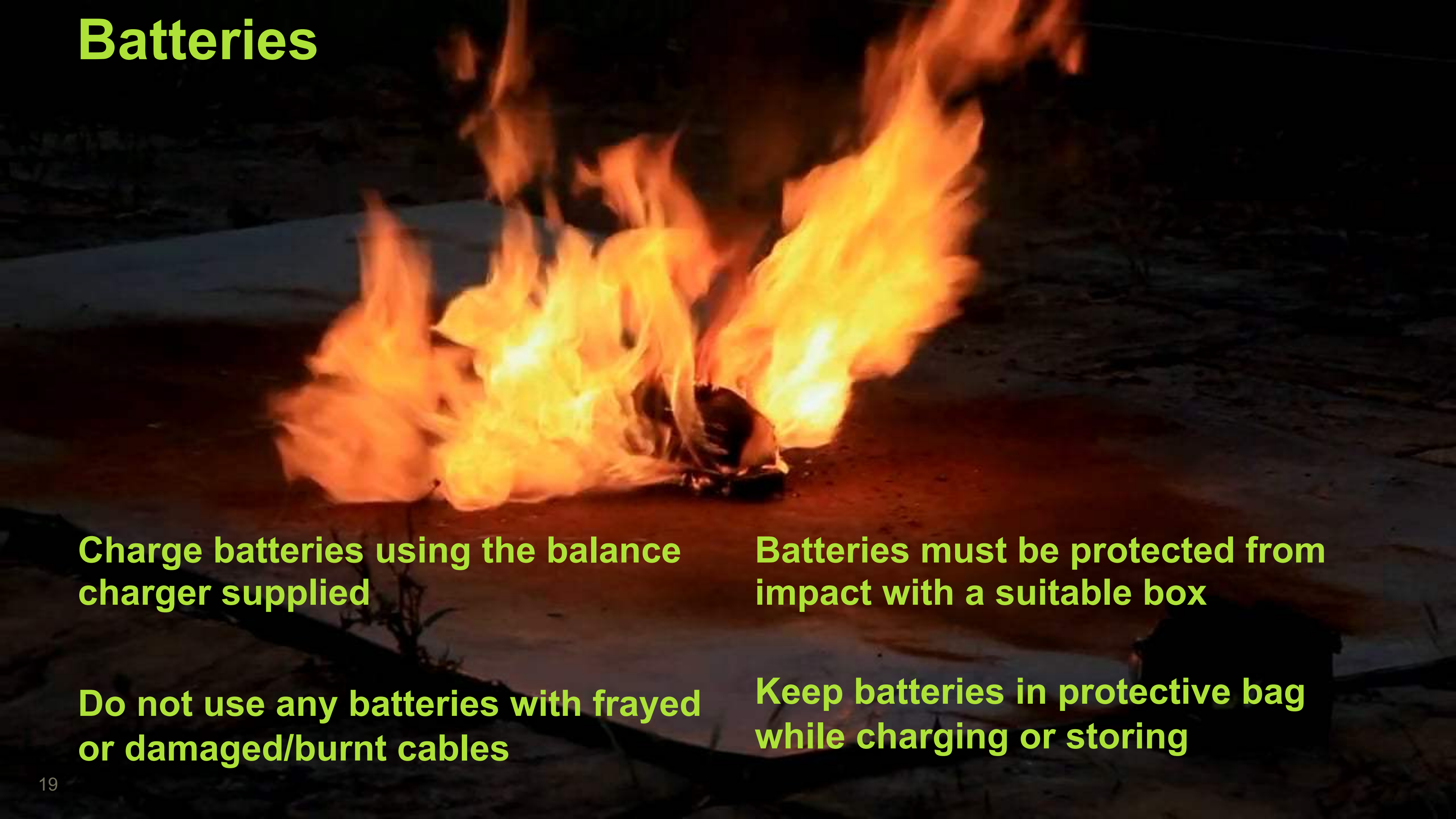
ROBOCON 2024

Q&A



The Robot Kit

Batteries



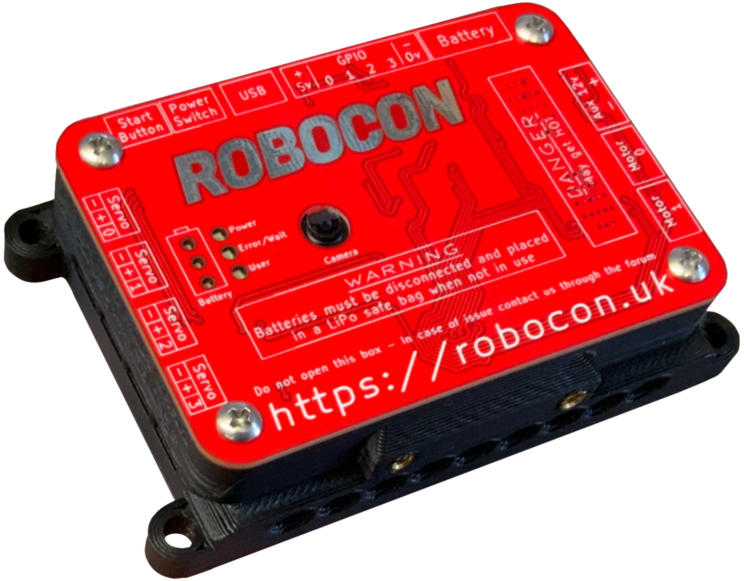
Charge batteries using the balance charger supplied

Do not use any batteries with frayed or damaged/burnt cables

Batteries must be protected from impact with a suitable box

Keep batteries in protective bag while charging or storing

Brain Box



The Brain Box contains:

- Raspberry PI
- Full HD Pi camera
- Battery protection system
- 4x PWM drivers for RC servos
- 4x Digital In/Out and Analog In
- 5v and 12v power for external devices
- WiFi access point for debug output and code upload
- Expansion available by USB

Windows PC

Windows PC Contains

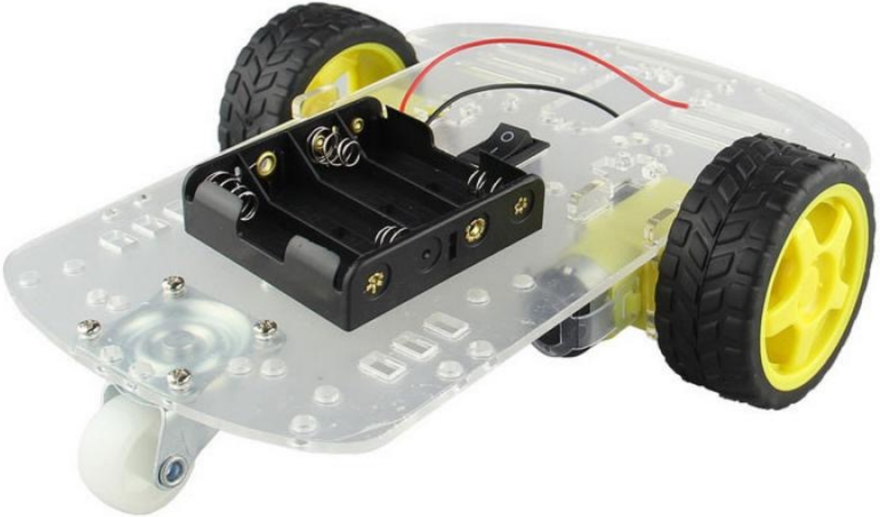
- Web Browser

- Use the web browser to upload robot & run code on the brain box

Mini-Bot

Basic robot chassis to get you off the ground

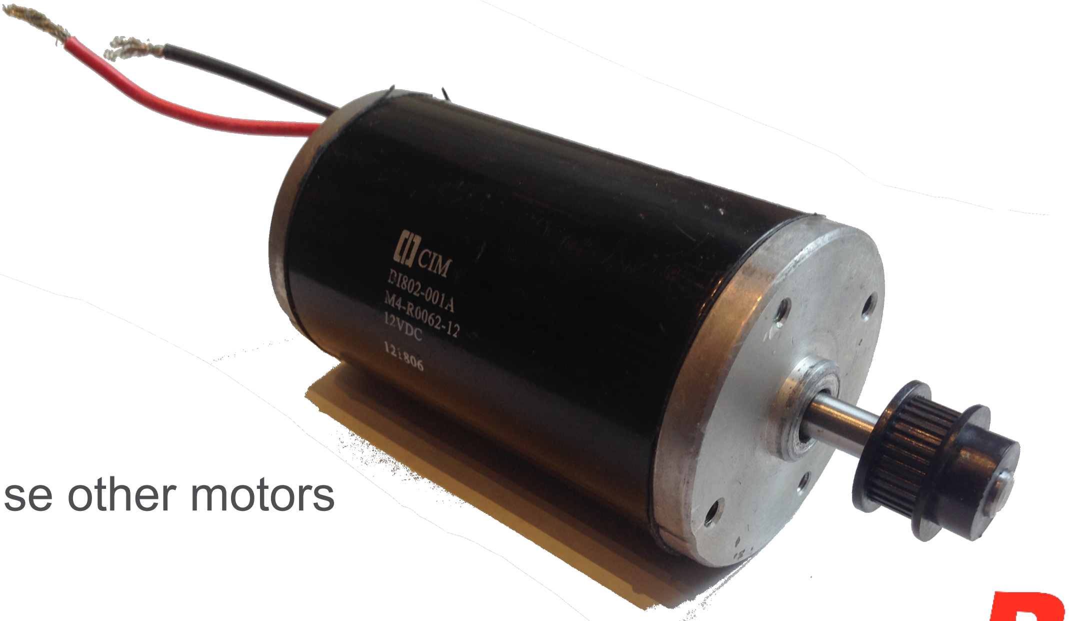
- Beware the Perspex!
- The motors are tiny



Note: Your mini-bot will differ slightly

DC Motors

The Brain box can directly drive Two DC motors



- Motors each have a 10A limit
- Set maximum motor power!

Note:

- DC Motor Stall Current
- Configure motor power if you use other motors

PWM outputs for Servos (1)

Up to four servos can be driven by PWM 1-4 on the Brain Box

Ground is the Upper pin

5v is the Middle pin

Signal is the Lower pin

Note:

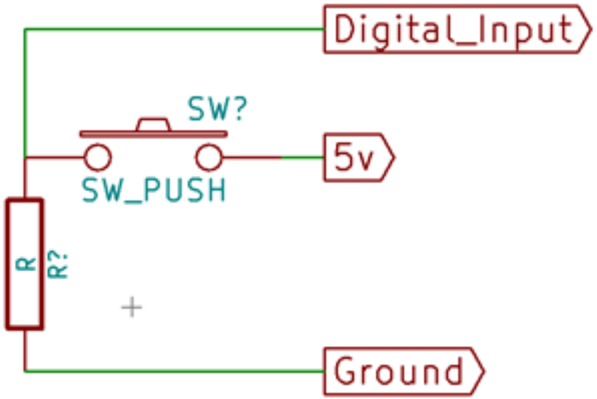
There are Five (!) color codes

- Red is standard for 5v
- Black is probably ground
  - Unless there is no black, then Brown is ground

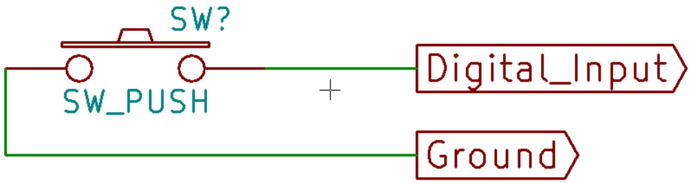
Digital Input

All the connections on the GPIO plug can be used as Digital Inputs

In Standard mode the input must be driven high or low



Using the “Pull Up” configuration switches can be directly wired



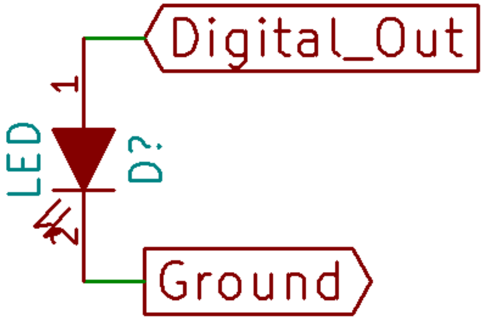
Note:

Switch logic is inverted using this method

Digital Output

All the connections on the GPIO plug can be used as Digital Outputs

Use the digital output directly for connecting Logic or LEDs



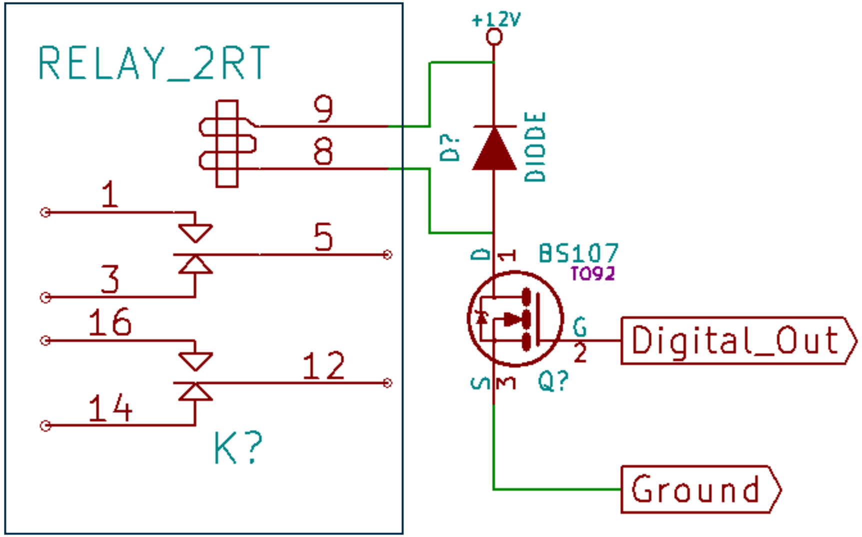
Notes

Integral 1k resistor limits current to 5mA, which may be quite dim with some LEDs

Digital Output (2)

High current devices such as Pumps can be operated using a relay

Use MOSFETs or BJTs to drive high current devices such as relays or high power LEDs



Notes:

Always use a diode to protect from back-EMF on inductive loads (like relays)

Using Analog Input

GPIO 1-4 are capable of reading Analog Input

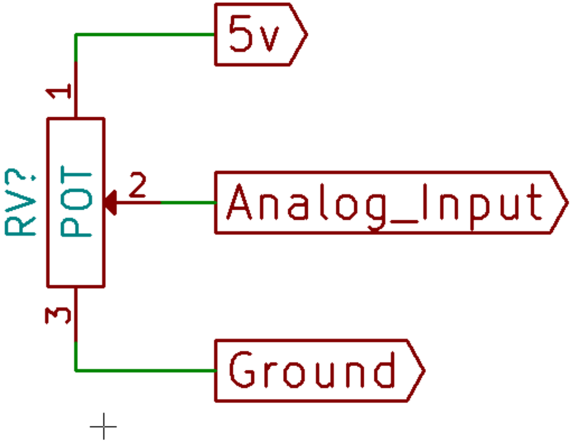
Note:

Variable resistors should be 10k Ohm or less for clean results

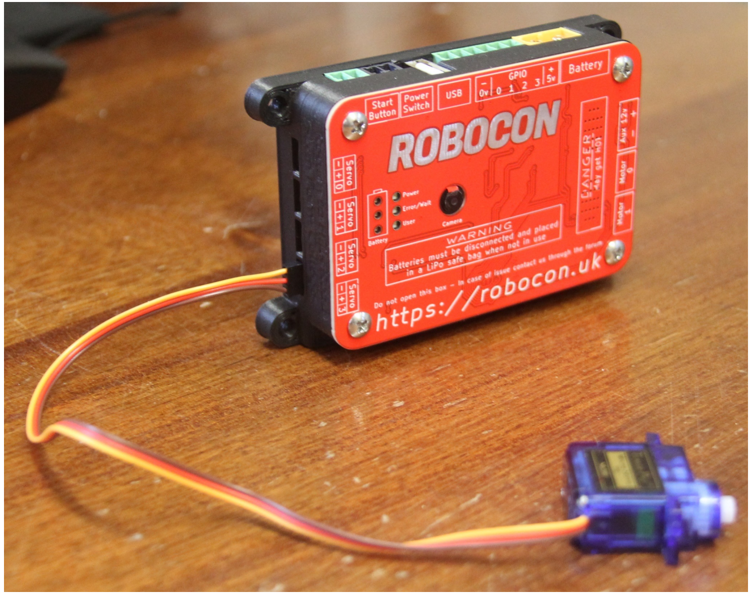
High impedance inputs may be affected by other inputs or motor current readings

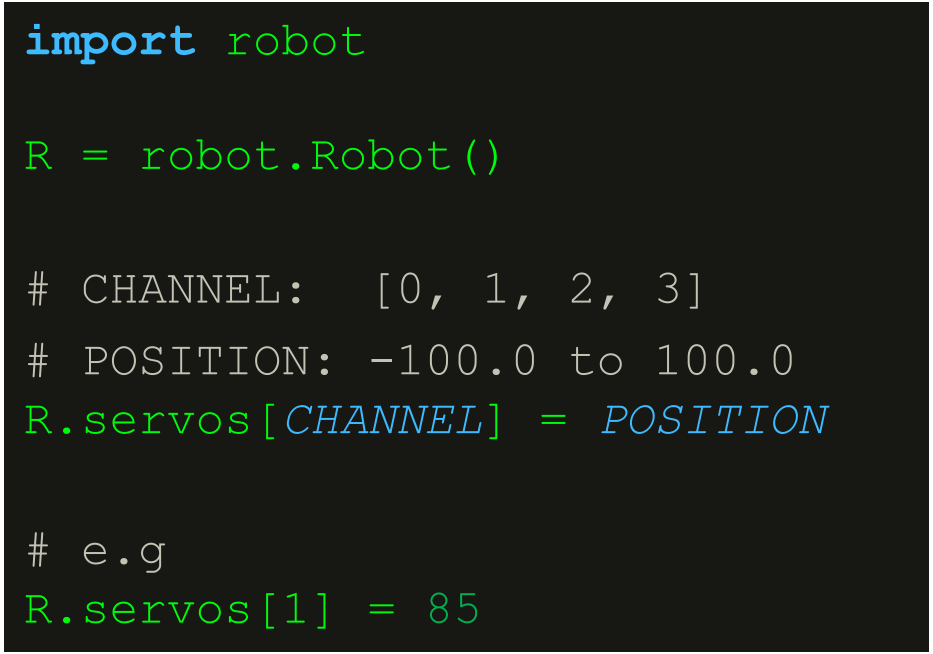
- Use an OP amp to buffer high impedance signals

Unconnected inputs may read as random values



PWM outputs for Servos



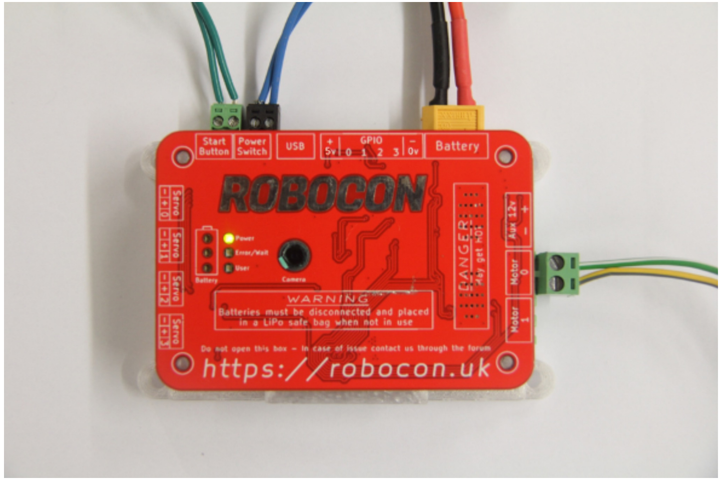


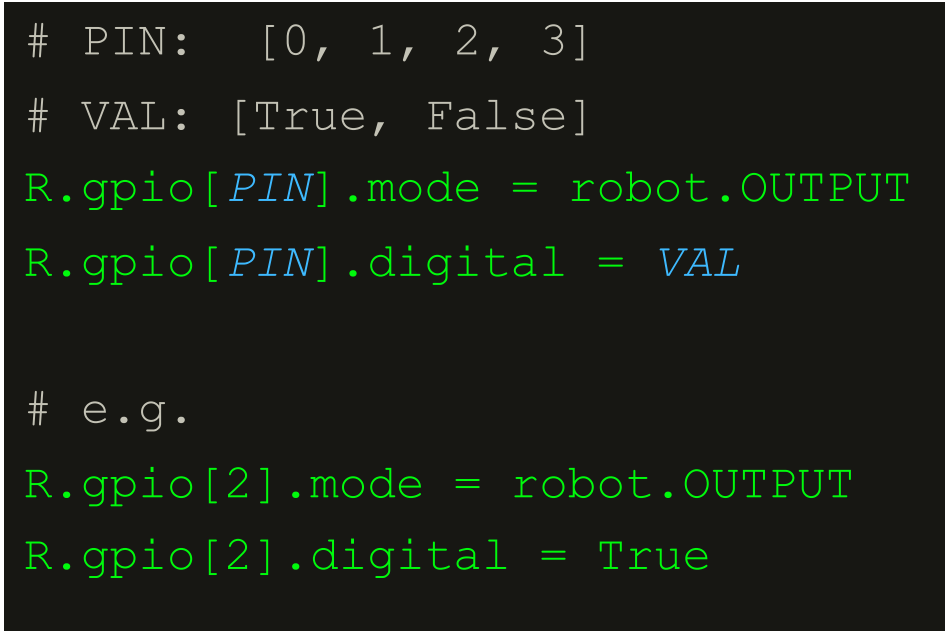
```
import robot

R = robot.Robot()


# CHANNEL:  [0, 1, 2, 3]
# POSITION: -100.0 to 100.0
R.servos[CHANNEL] = POSITION


# e.g
R.servos[1] = 85
```

Digital Input

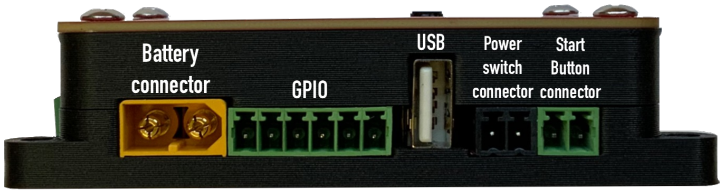
```
# PIN:   [0, 1, 2, 3]
# MODE: [INPUT, INPUT_PULLUP]
# Returns: True/False
R.gpio[PIN].mode = robot.MODE
print (R.gpio[PIN].digital)


# e.g.
R.gpio[3].mode = robot.INPUT
print R.gpio[3].digital
```

Digital Output

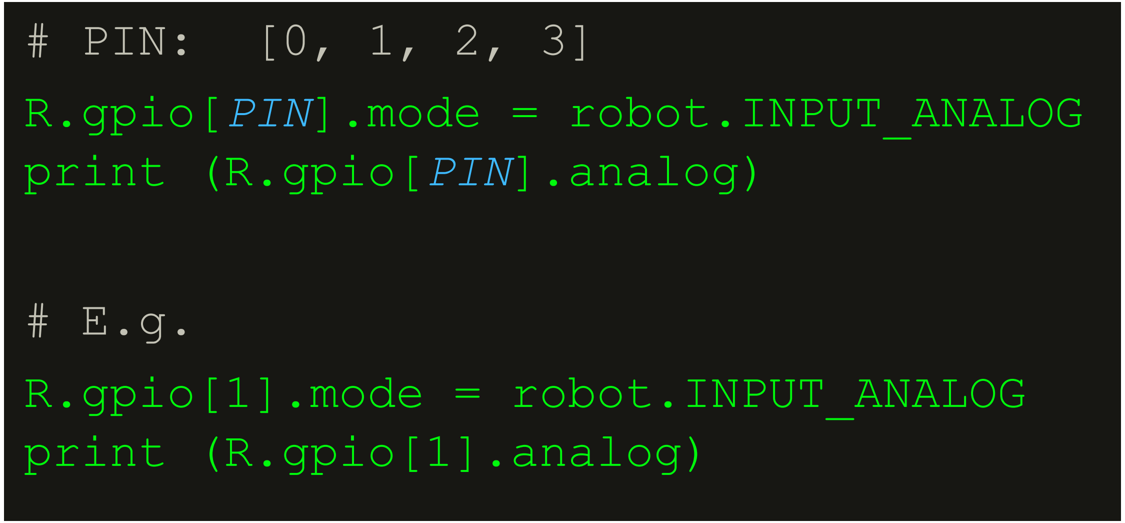

Start Button
Power Switch
USB
GPIO
5v 0 1 2 3 0v
Battery
ROBOCON
Servo
Power
Error/Wait
User
Camera
Aux 12v
Motor 0
Motor 1
DANGER
WARNING
Batteries must be disconnected and placed in a LiPo safe bag when not in use
Do not open this box - In case of issue contact us through the forum
https://robocon.uk



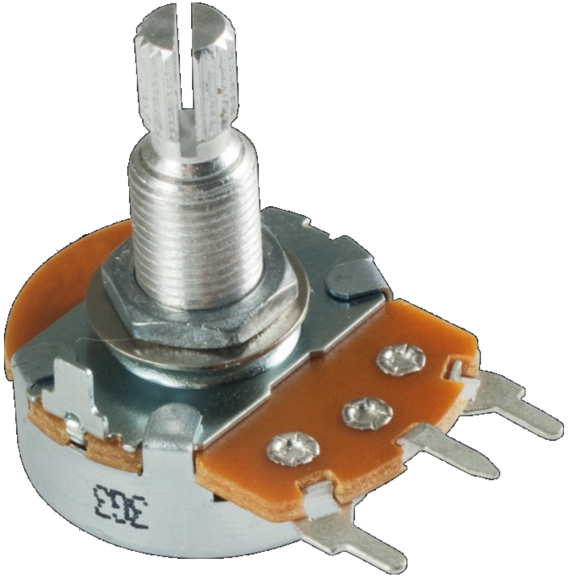
```
# PIN:  [0, 1, 2, 3]
# VAL: [True, False]
R.gpio[PIN].mode = robot.OUTPUT
R.gpio[PIN].digital = VAL


# e.g.
R.gpio[2].mode = robot.OUTPUT
R.gpio[2].digital = True
```



Analog Input



```
# PIN:  [0, 1, 2, 3]
R.gpio[PIN].mode = robot.INPUT_ANALOG
print (R.gpio[PIN].analog)


# E.g.
R.gpio[1].mode = robot.INPUT_ANALOG
print (R.gpio[1].analog)
```



DC Motors

```
R = robot.Robot(max_motor_voltage=9)


# POWER: -100.0 to 100.0
R.motors[1] = POWER
R.motors[2] = POWER


# e.g.
R.motors[1]= 100
R.motors[2]= -100
```

Support

- Email: robotics@hrsfc.ac.uk
- (Almost) Every Saturday we have a meeting
  - Check with us first
- Website:
  - Forums
  - Documentation

ROBOCON 2024

Final Q&A

What’s Next

Kit Pickup

Lunch

Workshop

CMR SURGICAL

redgate

FRONTIER

Microsoft

Microsoft is a registered trademark of the Microsoft Corporation