enrichment
tech

Tiger Enrichment Manual

Table of Contents

Maintanance

Troubleshoot

Beyond the Prototype

- Sustainability

Behind the Claw

- Development Iterations
- Design Challenges
- Team

Introduction

Big Cats Story

Tigers are amongst the most commonly known members of the big cat family. The Malayan tigers in particular typically reside in forested areas in swamps and scrubland. These carnivorous creatures are excellent predators; they hunt their prey using their keen sense of smell and sight. They can grow up to 140 kg and feed on 8 to 10kg of meat - mainly hoofed animals such as deer - per meal. Their typical lifespan in the wild is between 8 to 10 years.

Night Safari currently houses 4 captive Malayan tigers, 2 males and 2 females, whose welfare are cared for by a team of dedicated keepers. Current enrichment devices include Hessian cloth bags enclosing meat chunks that the tigers have to tear through, and introducing new scents in the form of grass planted in their exhibits. These engage them cognitively and encourage them to explore and to mark their scent by rubbing their body against the grass patches. However, these enrichments which are easily destroyed, lack success in offering a sustainable solution and oftentimes require physical labour that may compromise keepers' time and safety.

Objectives

Thus, this project proposes a robotic enrichment device, Dolly the claw, that can be implemented for the long-term in their exhibits.

- Encourage natural predatory behaviour: hunting and stalking

At present, zookeepers throw the tigers' food into their exhibit through holes in the fence measuring 20cm by 20cm. This way, the tigers

receive food passively as they do not need to search for their food. They may even develop pacing behaviour when they anticipate feeding time at the sight of keepers. By using the Dolly, we would like to encourage a feeding that requires tigers' participation prior to them getting the meat. The device allows zookeepers to control movement along the zipline to pique the interest of the tigers into engagement and hunting as they would in the wild, before fed the meat from the claw release as reward.

- Increase physical activity of tigers

Feedback from the keepers revealed problems that predatory animals such as the tigers tend to face: muscle degeneration from prolonged sedentary lifestyle in zoos. Our Dolly, thus, hope to encourage tigers to leap upwards towards the target as a form physical workout to maintain their muscular strength especially in the hind legs. As the release of the food is controlled by the zookeepers, they can control the trolley motion as much as they like to train the tigers, and only release the meat after the tigers have engaged in sufficient physical activity.

Dolly the Claw

Materials

/HARDWARE

- 150 x sets of (a) nuts, (b) bolts and (c) washers

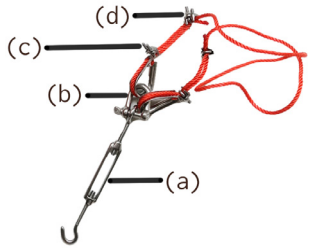
- Zipties

REMOTE

- 2 x (Muji) container 11cm x 7.5cm x 4.5cm
- 1 x (Muji) container 17cm x 5cm x 2cm

ZIPLINE

- 1 x 15m polyester rope
- 2 x sets of (a) turnbuckles, (b) shackles, (c) rope thimbles, (d) clips



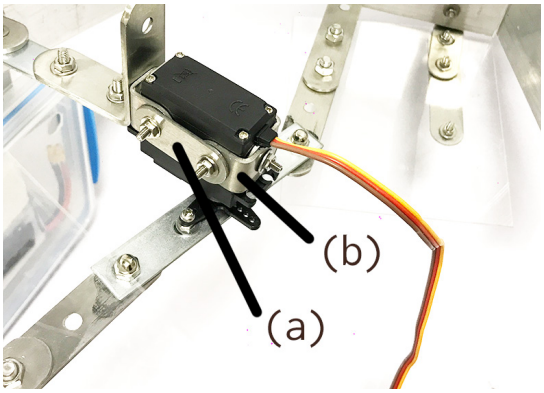
TROLLEY

- 1 x Aluminium C channel, 46cm x 7.5cm
- 1 x Aluminium C channel, 15.5cm x 7.5cm
- 4 x Stainless steel rigid castor wheels, diameter 5.3cm x 2cm
- 1 x L bracket with 4 holes, 5cm by 5cm x 1.7cm
- 1 x Straight bracket with 4 holes, 10cm x 1.5cm
- 1 x 3D-printed pulley wheel, diameter 6cm x 2.3cm
- 1 x Small (Lock and Lock) container 13.5cm x 10.2cm x 5.2cm

- 2 x Big (Sistema) Container 14.9cm x 15.4cm x 16.3cm
- 6 x Small L brackets with 2 holes, 2cm per side x 1.7cm
- 6 x Cable retainers with 2 hooks, 2.5cm x 0.7cm

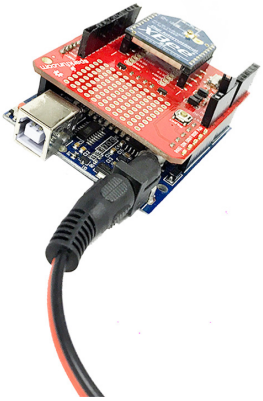
CLAW

- 2 x Hinges with 2 holes, 5cm x 1.5cm per side
- 2 x Metal plates 14.5 cm x 10 cm x 0.2cm
- 2 x Metal plates 17 cm x 10 cm x 0.2cm
- 2 x Metal plates 12 cm x 10 cm x 0.2cm
- 13 x L brackets with 4 holes, 4cm x 4cm x 1.7cm
- 5 x Straight brackets with 4 holes, 7.5cm x 1.7cm
- 2 x Anti-slip rubber 10 cm x 4 cm
- 4 x Acrylic sheets 11cm x 11 cm
- 1x set of (a) straight brackets with 2 holes, 3cm x 1.7cm, (b) small L brackets with 2 holes (1.5 cm by 1.5 cm x 1.7cm) (Servo holder)



/ELECTRONICS

- 2 x sets of XBee S1 modules, XVee shields, Arduino UNO, battery adapter to barrel jack (XBee Set)



- 2 x USB cables for Arduino UNO
- 3 x sets of 1.5V AA batteries, battery holders

- 1 x Velcro strips 50cm
- 1 x Shrink wrap tubing 50cm
- Jumper cables

DEVICE

- 1 x Worm gear right-angle reversible motor DC, 12V 20 rpm 20 kgcm torque
- 1 x HS - 311 Servo
- 1 x Li-Po Battery 11.1V
- 1 x H-Bridge SN754410NE
- 1 x Perfboard
- Male header pins

REMOTE

- 1 x PS2 Joystick two axis module

/TOOLS

- Electric drill
- Dremel
- Hot glue gun
- Solder
- Li-Po Battery charger
- Screw driver
- Wire cutter
- Wire stripper

How to: Remote

/ELECTRONICS

1. Connect 4 jumper wires to a XBee set: GRD, +5V, A5 and A4.

2. Load the sender code onto Arduino UNO of the above XBee set using a USB cable for Arduino UNO:

```
#include “SoftwareSerial.h”

SoftwareSerial XBee(2, 3);     // initializing XBee RX, TX Pins


int joyPinX = 4;         // slider connected to analog pin 4; control zipline

int joyPinY = 5;          // slider connected to analog pin 5; control claw


int valueA = 0;              // to read the valueY from analog pin 5

int valueB = 0;              // to read the valueX from the analog pin 4


bool clawIsOpen = false;        // track claw state

bool motorIsFwd = false;        // track motor direction

bool autoison = false;          // track auto mode


int treatValue(int data) {       // joystick value conversion

  return (data * 9 / 1024) + 48;

}


void setup()

{

   Serial.begin(9600);

   XBee.begin(9600);

}
```

```
// to read debug transmission
void loop2() {
  if(XBee.available()) {
    char a = XBee.read();
    Serial.println("a");
    // check for LED blink on shield
  }
}

void loop()
{
  valueA = analogRead(joyPinY);
  valueB = analogRead(joyPinX);

  if(valueA < 100 && clawIsOpen){
    //when pushing up
    Serial.println(valueA);
    Serial.println("Open Claw");
    XBee.write("<A>");
    clawIsOpen = false;
  }
  else if (valueA > 300 && valueA < 520 &&
!clawIsOpen){
  // when button at neutral
    Serial.println(valueA);
    Serial.println("Close Claw");
    XBee.write("<B>");
    clawIsOpen = true;
  }
  delay(20);

  if(valueB < 100 && motorIsFwd){
    Serial.println(valueB);
    Serial.println("Move Direction A");
```

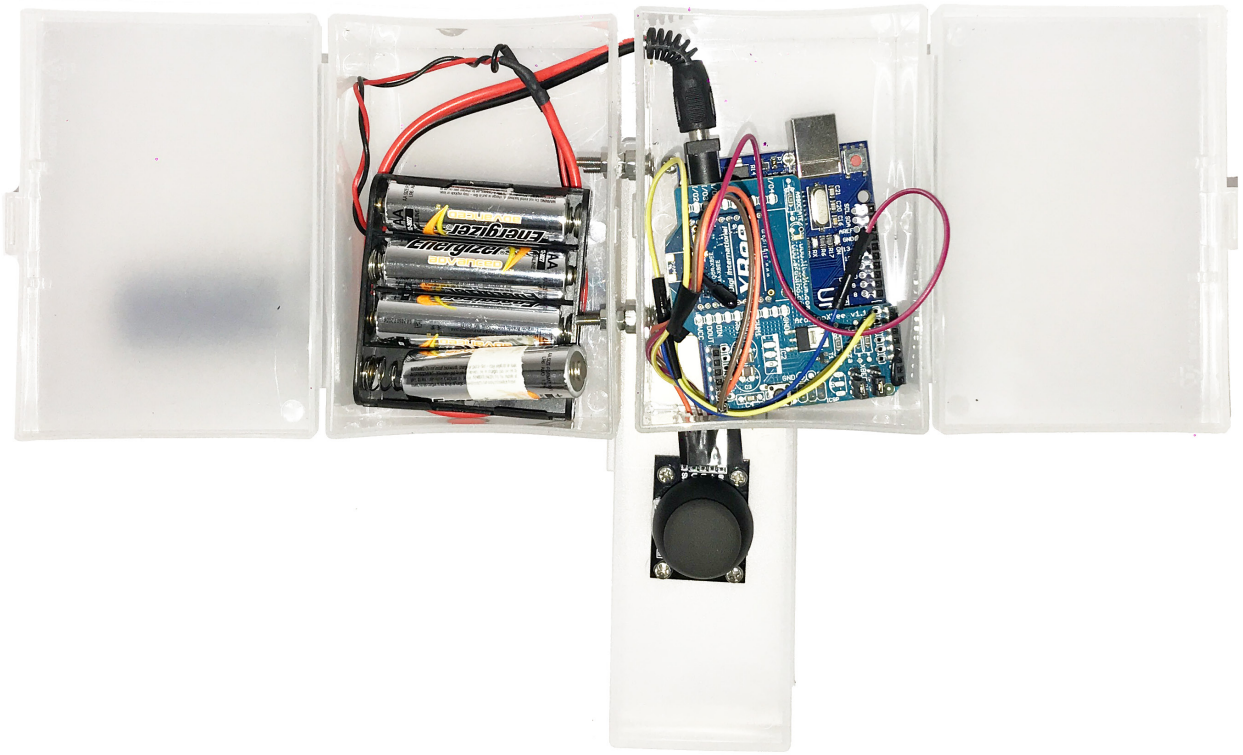
```
  XBee.write("<C>");
  motorIsFwd = false;
}
else if (valueB > 700 && !motorIsFwd){
  Serial.println(valueB);
  Serial.println("Move Direction B");
  XBee.write("<D>");
  motorIsFwd = true;
}
delay(20);

// turning auto zipline on
if (valueA > 700 && !autoison){
  Serial.println(valueA);
  Serial.println("Auto On");
  XBee.write("<E>");
  autoison = true;
  }
  else if(valueA < 700){
   autoison = false;
  }
  delay(20);

}
// end
```

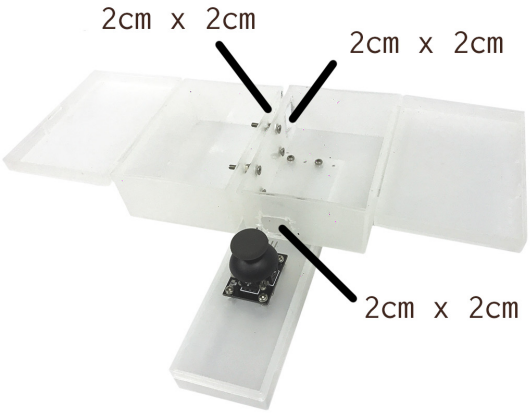
3. Solder the wires of the battery holder to the battery adapter, Black to Black for ground and Red to Red for power.

4. Place shrink wrap tubing around the soldered area and apply heat to protect it.

/HARDWARE

1. Drill 2cm x 2cm squares in the 3 (Muji) boxes using an electric drill and screw them together as follows:





2. Drill the joystick on top of the bottom box and in front of the 2cm x 2cm square as shown.

3. Connect the battery adapter to the barrel jack of the Arduino UNO in the

4. Connect the 4 jumper wires attached to the XBee set as follows:

   GRD to GRD

   +5V to +5V

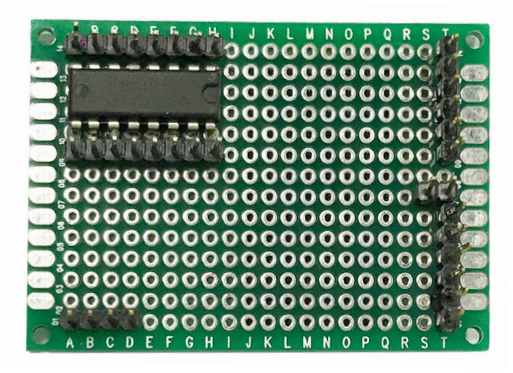
   VRx to A5

   VRy to A4

How to: Device

/ELECTRONICS

1. Solder the H-bridge and a few rows of male header pins to the perfboard as shown:



2. Connect a perfboard, a XBee set, a motor and a servo as follows:

- Connect GND of Arduino Xbee Shield to one row of male header pins to create a ground rail:
  i) Connect to ground wires of Servo, Battery holder, Pin 4 (or 5) of H-bridge.
  ii) Attach an additional long wire to ground rail to be connected to ground of 11V LiPo battery pack

- Connect 5V from Arduino Xbee Shield to the another row of male header pins on to create 5V power rail:
  i) Attach middle wire of servo, Pin 1 and Pin 8 of the H-bridge

- Attach a long wire to VCC (Pin 16) of H-bridge. This wire will be used for powering the DC motor after being connected to the 11V LiPo live wire.

3. Connect Xbee Shield to a H-bridge using jumper wires:

Arduino Pin 8 to H-bridge Pin 2

Arduino Pin 9 to H-bridge Pin 7

4. Attach two jumper wires from Pin 3 and Pin 6 of H-bridge. These wires will output voltage into the DC motor.

5. Attach live wire of Battery Holder (4x 1.5V) to a third set of header pins as shown to create another power rail solely for the servo motor.

- Connect live wire (orange) of servo to the same rail.

6. Connect the servo's signal pin to PWM pin 13 of Arduino Xbee Shield.

7. Load the receiver code onto the Arduino UNO of Xbee Board:

```
#include <SoftwareSerial.h>

SoftwareSerial xbee(2, 3); // initialize xbee RX,TX pins
#include <Servo.h>

Servo servo;


// motor chip digital pins

int enablePin = 11;

int in3Pin = 8; // receives logic from arduino to rotate clockwise

int in4Pin = 9; // receives logic from arduino to rotate anti


// autoloop variables

bool isRunningAutoLoop = false;

int autoLoopPhase = 0;
```

```
int autoLoopTimer = 0;

int lastAutoLoopUpdate = 0;



// main function variables

int motorSpeed = 0;

int angle = 50;

bool is_cw = true; // track motor direction



void setup()

{

  Serial.begin(9600);

  xbee.begin(9600);

  servo.attach(13); // servo logic pin attached to
PWM pin

  pinMode(in3Pin, OUTPUT);

  pinMode(in4Pin, OUTPUT);

  pinMode(enablePin, OUTPUT);

}



void loop()  {



 if(isRunningAutoLoop) {

  autoLoopTimer += millis() -
lastAutoLoopUpdate; //timer check

  lastAutoLoopUpdate = millis();

  Serial.println(autoLoopTimer);
```

```
  if(autoLoopPhase % 2 == 0) {
    //count even
    setMotor(255, true);
  } else {
    //count odd
    setMotor(255, false);
  }


  if(autoLoopTimer >= 3000) {
    autoLoopTimer = 0;
    autoLoopPhase ++;
    // when 6 clockwise and 6 anti-clockwise
rotations completed, auto mode stops
    if(autoLoopPhase >= 12) {
      autoLoopPhase = 0;
      isRunningAutoLoop = false;
      setMotor(0, true);
    }
  }
 } else {
   // when manual mode, just perform motor
from remote
   setMotor(motorSpeed, is_cw);
 }


 servo.write(angle);
```

```
// XBee transmission to xbee, signal conversion
String input = "";
while(xbee.available()){
  char x = xbee.read();
  if(x == '<') {
    input = "";
  } else if(x == '>') {
    handle(input);
    input = "";
    break;
  } else {
    input += x;
    }
  }
}
void handle(String input){

  Serial.println(input);

  //claw control
  // motor is immobile when performing claw function
  if(input == "A"){
    angle = 10;
```

```
  motorSpeed = 0;
  Serial.println("Claw Opens");
}

if (input == "B"){
  angle = 50;
  motorSpeed = 0;
  Serial.println("Claw Closes");
}

// driver control
if(input == "C"){
  motorSpeed = 255; // max speed
  is_cw = true;
} else if (input == "D"){
  motorSpeed = 255;
  is_cw = false;
}

// driver on auto control
if (input == "E" && !isRunningAutoLoop){
  isRunningAutoLoop = true;
  lastAutoLoopUpdate = millis();
 }

}
```

```
// motor function controlling motor output and direction

void setMotor(int motorSpeed, boolean reverse)

{

 //Clockwise

 if(reverse){

  digitalWrite(in3Pin, LOW);

  digitalWrite(in4Pin, HIGH);

 }

 if(!reverse){

  digitalWrite(in3Pin, HIGH);

  digitalWrite(in4Pin, LOW);

 }

 if(motorSpeed==0){

  digitalWrite(in3Pin, LOW);

  digitalWrite(in4Pin, LOW);

 }

}
```

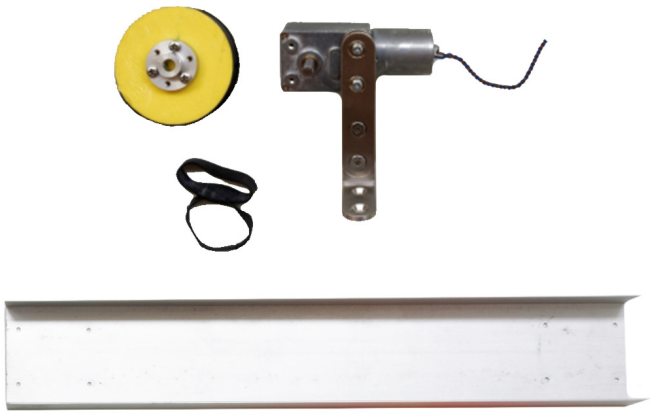
8. Solder the wires of the battery holder to the battery adapter, Black to Black for ground and Red to Red for power.

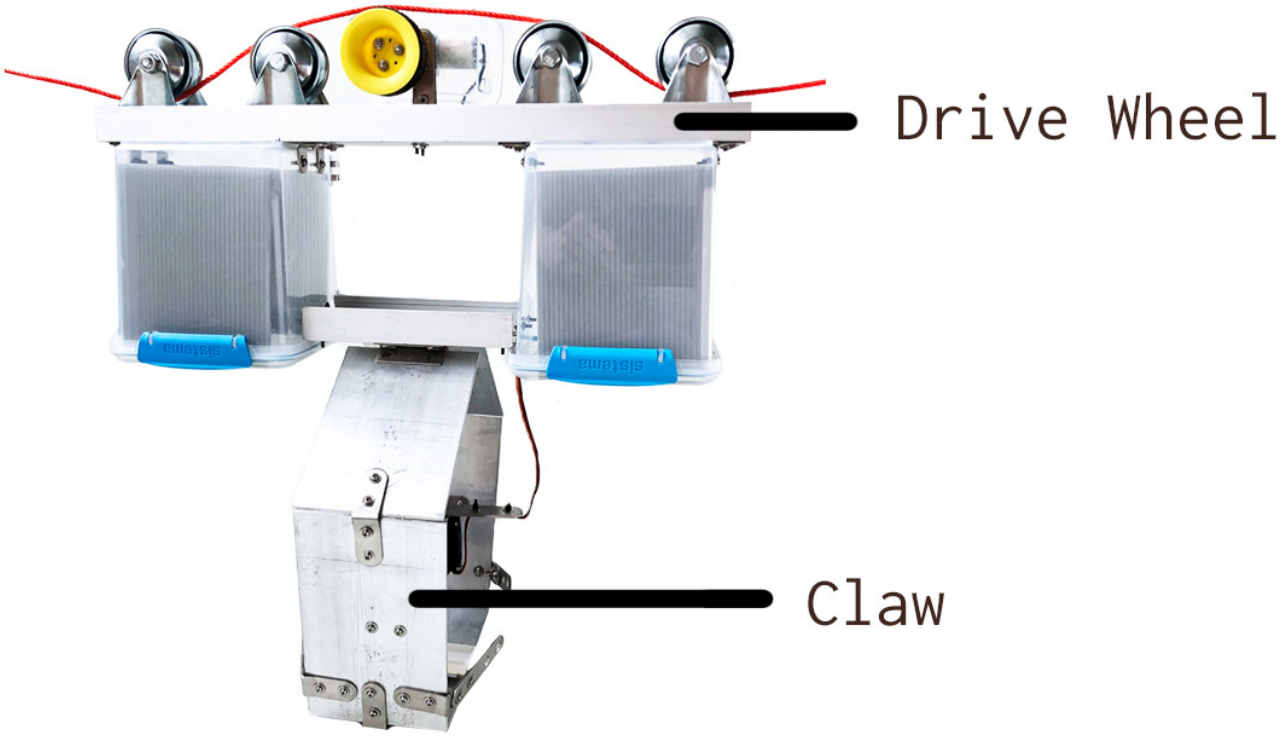
9. Place shrink wrap tubing around the soldered area and apply heat to protect it.

/HARDWARE

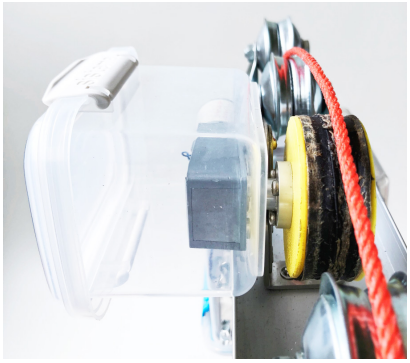
1. Drive wheel assembly: Start with a sturdy

piece of material as the platform for the drive wheel assembly at the uppermost layer of the device. We used a piece of C-channel aluminium 46.0 cm x 7.5 cm x 2 mm thick. The material is easier to cut and drill than stainless steel, while retaining a good balance between weight and strength. The channel ribs prevented much flex.

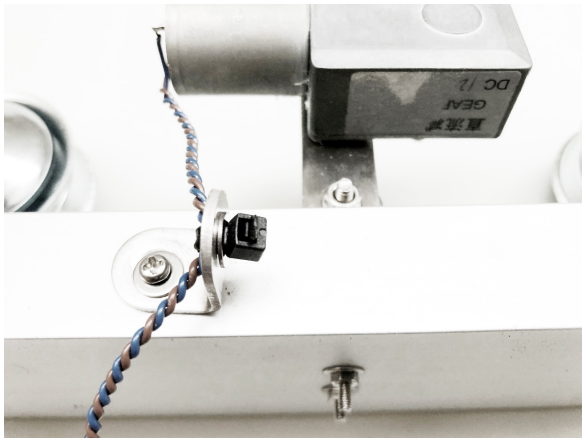




- Mount one drive wheel and drive wheel motor in the top middle surface of the C channel platform. We attached a custom 3D-printed drive wheel with a Pololu #2693 8mm aluminium universal mounting hub directly to a right-angle reversible worm gear motor with a corresponding 8 mm D shaft, then mounted the motor on to the platform with a stainless steel L bracket and straight mending bracket.
  i) Use a motor with sufficient torque and speed. The physics are such that the tension in a zip line imposes tremendous down-force on the drive wheel. The drive wheel motor has to have sufficient torque to overcome this resistance and sufficient turning speed (rpm) to move the trolley along the zip line at a reasonable pace. We used a no-brand one at DC 12V, 20 rpm, 20 kgcm torque.
  ii) Protect the drive wheel motor in a weather-resistant housing. We used a plastic food box container with a locking lid and cut holes for the motor shaft and cables.



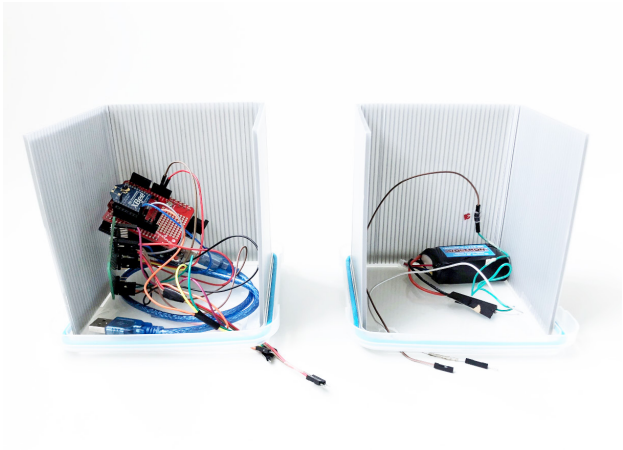
- iii) Use L brackets and zip ties to secure motor cables as anti-jerk protection.



- Mount two support wheels on either side of the drive wheel. Use smooth “pulley”-style wheels with mounting brackets for support wheels. We used rigid ball-bearing castor wheels with brackets, with the rubber tyres cut off to expose to the concave rim.

2. Housing for electronics assembly: Mount two containers to the underside of the drive wheel platform to serve as weather-resistant housing for the electronics assembly.

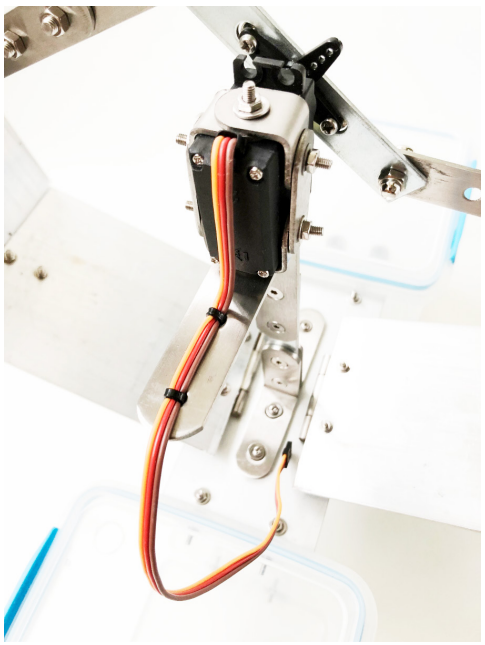
- We used two large plastic food box containers with locking lids, mounting one each with L brackets beneath the two pairs of support wheels. We inverted the containers so the mouths faced downwards, so that the electronics can be placed on the lids and pushed up into the housing and locked.



3. Claw assembly: Start with a sturdy piece of material as the platform for the claw assembly at the bottom most layer of the device. We used the same aluminium C channel 15.5 cm x 7.5 cm x 2 mm thick

- Attach the aluminium platform between and the bottom of the two electronics housing containers with L brackets.

- Mount the servo motor support structure using L and straight brackets to centre underside of the claw assembly platform. Mount the servo motor to this support structure, securing loose cables with zip-ties as anti-jerk protection.

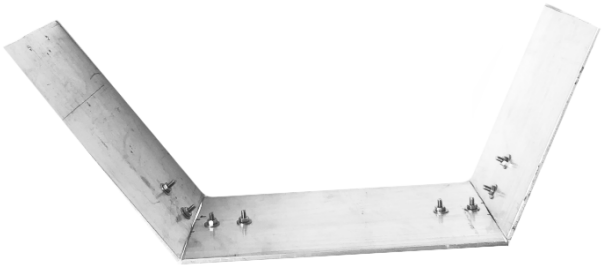


- Mount the two claws to stainless steel hinges on either side of the central support structure.

  I) We constructed the two claws out of a total of six pieces of 2 mm aluminium sheets cut to size: three pairs each of 14.5 cm x 10 cm, 17 cm x 10 cm, and 12 cm x 10 cm. We secured the plates to each other with L brackets bent to the desired angle with clamps and pliers.

  II) Attach plastic side plates to the claws with long L brackets bent to the desired angle with clamps and pliers. We used clear 2mm acrylic sheets cut to size. This serves to further secure the payload within the claw cradle.

  III) Attach anti-slip rubber pads to the claw ends to increase grip. We used four small door wedges.



4. Attach the pivot and lever drive mechanism from the servo motors to the claws using straight and L brackets. The pivoting joints of this mechanism have to be loose, so that they transform the back and forth rotating action of the servo shaft via the straight bracket levers into the hinged action of the claws. We used bolts smaller than the holes in the brackets secured with end caps as pivots.

Assembly and Use

When you are ready to use the device,

1. Assemble the trolley to the claw and the

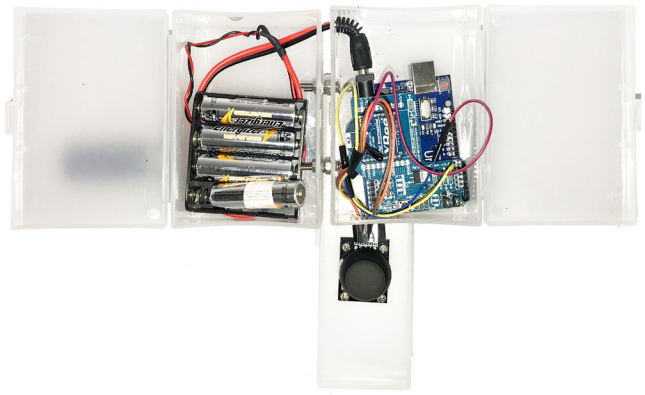
remote

2. Run the zipline through the pulleys as shown:

3. Secure each end of the zipline to a sturdy structure such as a tree, lamp post or railing.

4. Insert 4 AA batteries into the 3 battery holders to start up the devices.

/CONTROL



Orientation: Remote control facing upright

- Up (open claw, stops motor motion), release push to close claw
- Left (move left)
- Right (move (right)
- Down (auto loop)
- Motor function continues after pushing left or right once, until joystick is pushed 'Up' to stop motor.

Precaution

Always make sure that the live and ground wires of both the 11V LiPo battery and Servo motor (when connected) never touch each other, to prevent short circuiting and potential combustion of the machine.

You could:

- Use a battery pack adapter head so that the pins are insulated from one another
- Use female header pinheads as temporary

insulation covers

- In cases when wire rubbers have worn off, use duct tape as quick fix (but only temporary), replace wires as soon as possible.

Troubleshoot

- Check if the batteries are depleted of power, or leaking. Change if necessary.
  i) If the Arduinos are powered, the PWR (power) led bulb on the Arduino boards will be lit in red.

- Check the wires visually if there are any exposed or damaged areas, and that all wires are fully plugged-in in remote and claw.

- Check if XBee shields are working (i.e. configured and transmitting), both shields should show a green light blinking.

- Identifying your XBees:
  i) Remote XBee shield - Blue Arduino Libelium XBee Shield
  ii) Trolley XBee shield - Red Sparkfun XBee Shield

/REMOTE

- Check for broken antennas of the XBees modules and depleted battery pack.

- Wireless Communication Test:
  i) Toggle on any direction of the joystick: Each time an action is performed on joystick module, DOUT pin of Red Sparkfun XBee shield of your receiver Arduino would blink green, signal is sent successfully by remote, and received.
  ii) Check for DI05 LED light on the Red Sparkfun shield blinking green: Signal is transmitting
  iii) Troubleshoot:
  a) Perform a reset (tiny red button on Arduino board)
  b) Repeat Communication Test

c) Else, replace with new XBee module as last resort

- If Communication Test succeeds, test functionality of joystick module
  i) Reload sender code onto your Arduino board.
  ii) Check for 'jumper' settings (the small removable plastic sleeves that each fit onto two of the three pins labelled Xbee/USB). The sleeves should be fit towards in 'USB' position when uploading code.
  iii) Reset 'jumper' settings to 'XBee' after uploading.

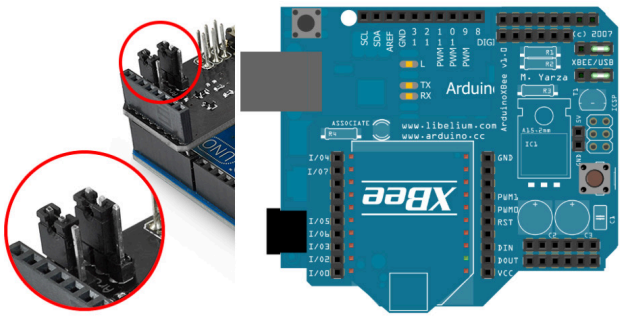


Image source: cooking-hacks.com (left), laserbots.blogspot.sg (right)

iv) Open Serial Monitor (Ctrl + Shift + M) and try toggling the joystick
v) If working, commands should be printed, such as "Open Claw", "Move Direction A", "Auto Mode" etc.

/DEVICE

- Carry out simple communication test (see above on General Troubleshoot). Else, reload receiver code onto Arduino board:
  i) Open Serial Monitor (Ctrl + Shift + M),
  if transmission from remote is successful, commands should be printed "A Open Claw", "B Close Claw" etc.
  ii) Ensure that switch in diagram below (on the XBee shield) is in DLINE position so that configuration of transmission pins is not interrupted.

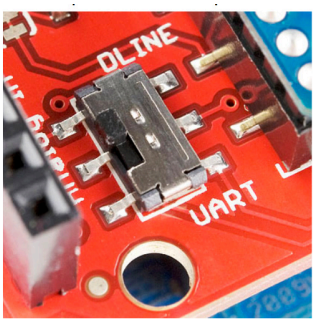


Image source: learn.sparkfun.com

/CLAW

- If communication test successful and the claw is not opening or closing when toggling remote, do the following:
  i) Ensure that the wires to the claw are correctly wired. (Refer to shield hookup guide above)
  ii) Ensure there are no loose wires in the electronics casing.
  iii) Check that there are no loose bolts on the claw
  iv) Check if battery powering servo is depleted. Look out for servo motor 'pulsing', or only twitching slightly when remote toggled

/TROLLEY

- Test for XBee transmission following General Troubleshoot above.
- If Xbee transmission is working but the trolley is not moving when moved left or right (on the remote control), do the following checks on your DC motor wiring and hardware:
  i) Ensure battery wires are wired correctly
  ii) Ensure 11V battery is charged.
  iii) Ensure no loose wires in the electronics casing.
  iv) Check that the motor is not faulty by attaching the 11V battery pack directly to the motor. If the wheel is able to turn, the motor is still working. Otherwise, replace the motor as it is faulty.
  v) Check the rubber on the wheel. If it has worn out, replace it and test the trolley on the zipline again.
-

Beyond the Prototype

Sustainability

Moving forward, WRS can consult with engineers to create industrial-grade infrastructure to be permanently fixed above the big cats enclosure. This ensures the safety and durability of the enrichment device that would last.

Our current zipline model can be further designed to encourage tigers' natural pouncing abilities during hunt by constructing robust claws to withstand tigers' weight (up to 150kg) if they were to attempt to grab its food from the device hanging above.

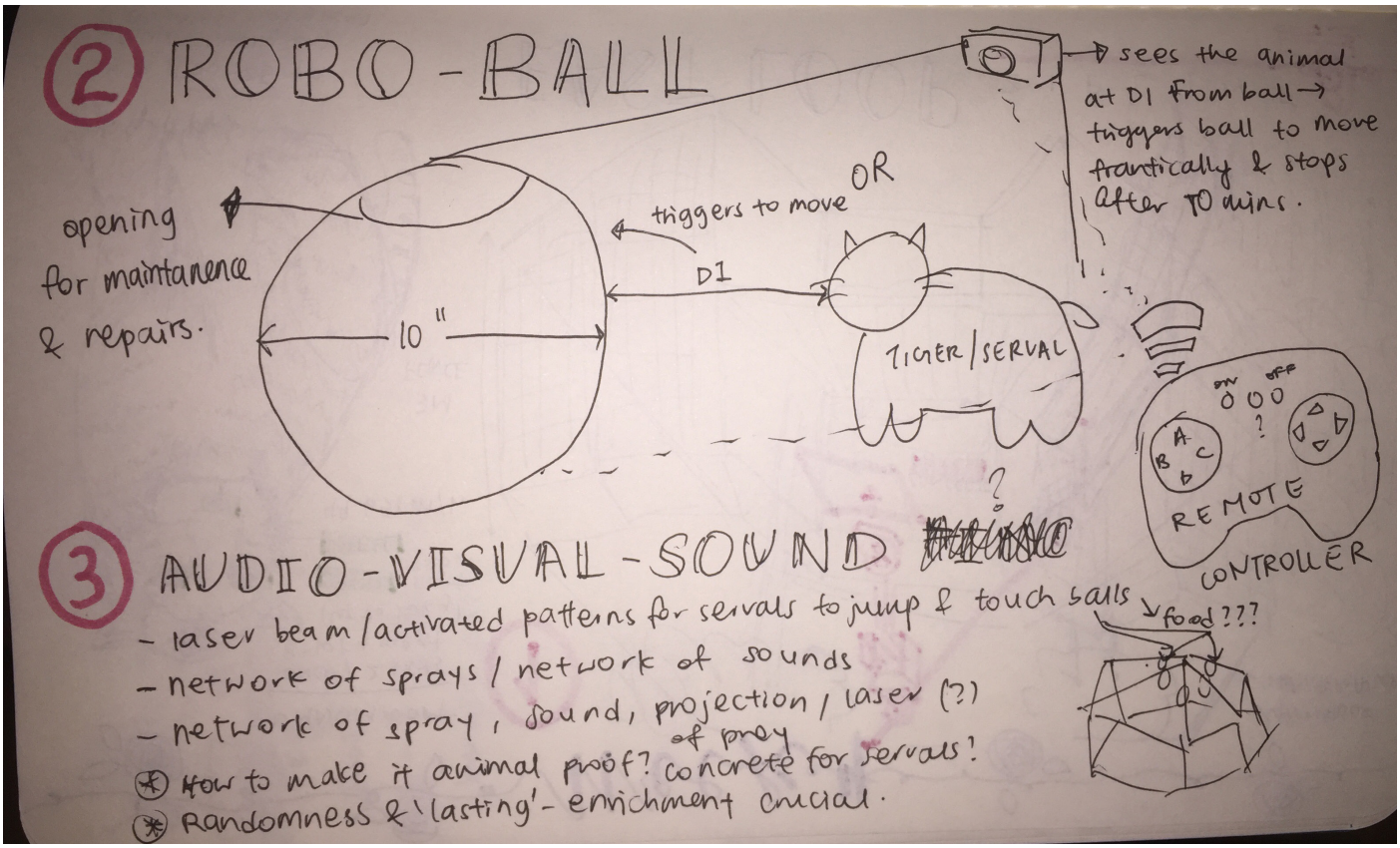
Alternative components for the electronics can be explored. Smaller and more compact remote controls can be designed for the convenience of zoo keepers who will carry them around. Otherwise, mobile phones with a compatible application that functions on bluetooth connectivity can be looked into further. This compacts the remote control into a single tap on keepers' mobile phone, which they currently bring along with during work.

Given the mobility of the device and remote controls, using Xbee modules with trace or UFL chip antennas rather than those with wired antennas will make the claw more resilient, for the latter is more prone to damage.

Behind the Claw

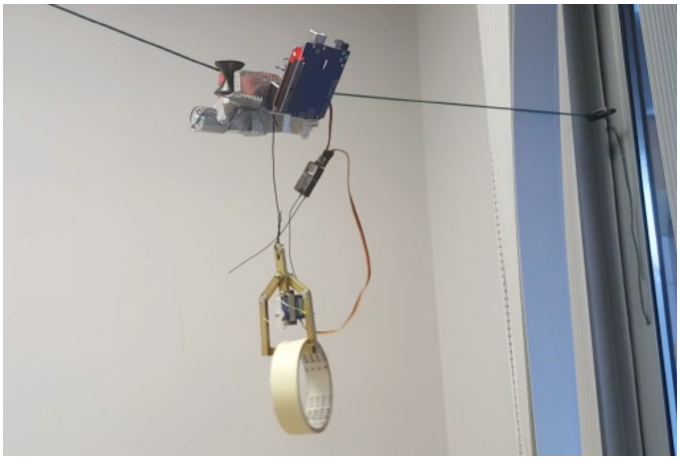
Design Iterations

/INITIAL IDEAS

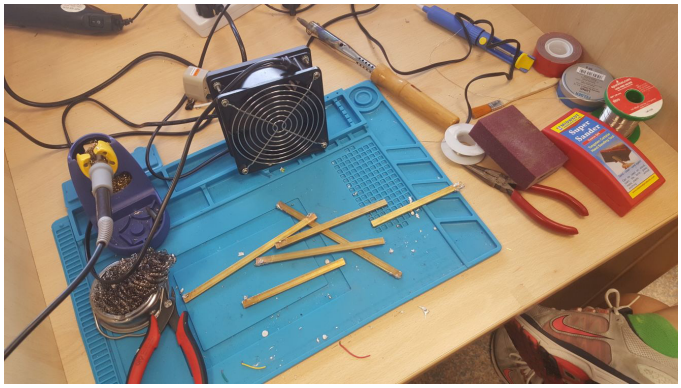


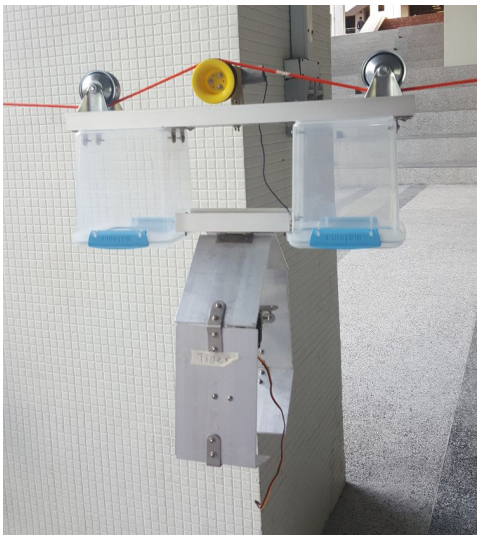
Initial ideas such as a robotic ball and audio-visual stimulation were suggested. While these ideas were interesting and plausible enrichment for the tigers, due to limited budget, the robotic ball is less than feasible; the keepers also brought up concerns about audio-visual stimuli that may encourage undesirable visitor behaviour (using laser pointers) and brightness from the visuals might disrupt zoo experience for visitors.

/LOW-FIDELITY PROTOTYPES



Beyond our initial prototype, we encountered many issues along the way: designing a robust claw that not only withstand weights but is also weather and ant-proof; getting the angles right; selecting the right motor with the correct torque and hence speed; selecting the correct material for the rope; physics of the trolley to spread out weight across not 1 but 3 pulleys.

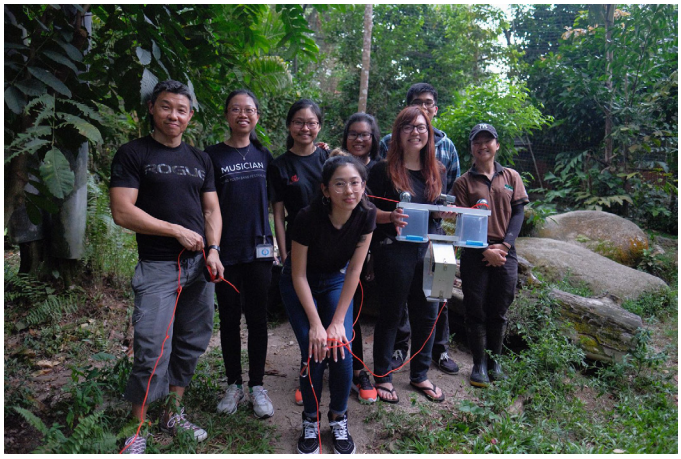




We also experimented with different remote designs: ball, round container. Eventually, we

settled on a design that has better grip and can work single-handedly. Buttons were chosen over joystick as well, which affords greater control.

We had to reconfigure the coding, even up until the day before our final group field trip to test our prototype with the tigers.



Design Challenges

First and foremost, no zookeepers can be in the same enclosure as the animals. This rendered the need for a wireless connection between the remote control and the device and presented a challenge to building our electronics.

Another challenge faced by our team is that we are unable to test our prototype in the tiger's exhibit or enclosure due to safety constraints. As a 5 kg device that should be set up 4m in height out of reach of the tigers and for the zookeepers to only be exposed as minimally as possible, proper safety precautions and infrastructure are required to protect the safety of both the tigers and the zookeepers. This meant that we were not able to test our prototype in the tiger's exhibit directly.

During one of our demonstrations, we attempted to test our device above the back-of-house enclosure of the tigers. Thus, we were unable to get an accurate reaction from the tigers as the device was not within the enclosure. Furthermore, no meat was used during the test.

Team

Ho Koon Yee Kaitlyn
Year 3, Communications and New Media major

Lian Hui Shan
Year 2, Communications and New Media major

Neo Yi Hui
Year 2, Communications and New Media major

Ng Yi Rou Isabel
Year 2, Communications and New Media major

Ngiam Ling Li Vivian
Year 2, Communications and New Media major

Yap Chuin Houi
Year 2, Industrial Design major

This project is a collaboration with the Wildlife Reserves Singapore under NM3231: Physical Interaction Design. For this project, we were tasked to come up with an enrichment device for the Malayan Tigers at the Night Safari.