ScraphEEp 2018

Challenge

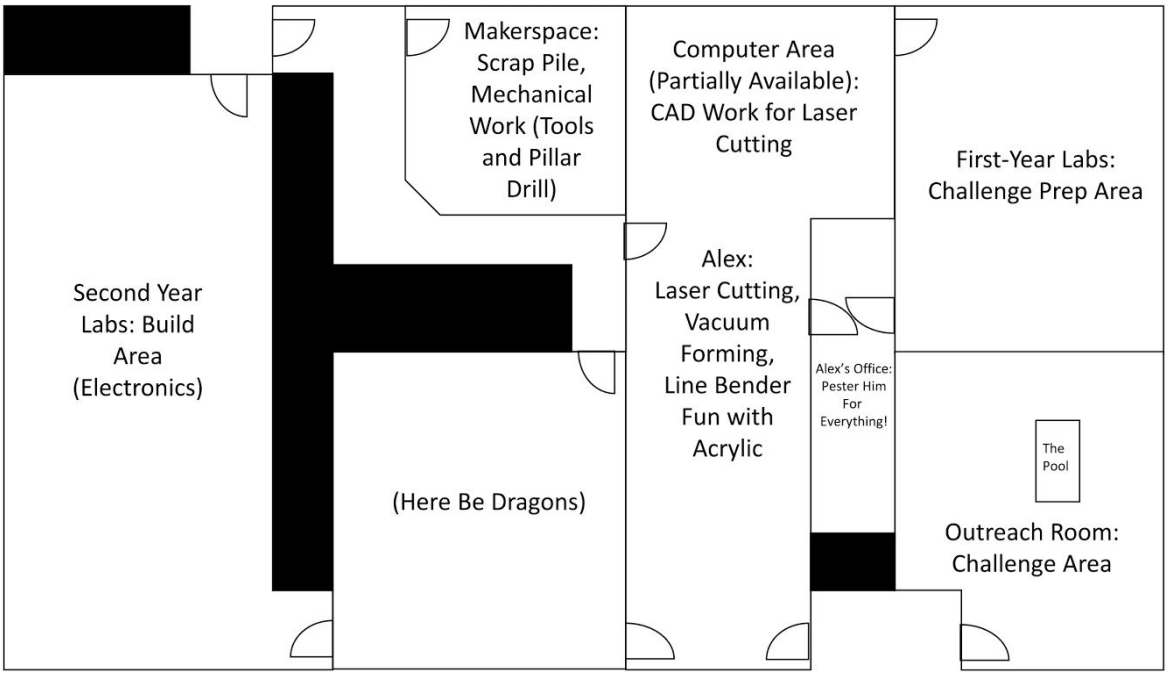
Build a WiFi-controlled contraption to collect the baby ducks in the EARS paddling pool™ and return them to their mother!
Each team has four hours to build a contraption to complete this task as quickly and as well as possible. Your contraption should be no bigger than a sink and must be able to float and function in (at most) 20cm of water.

Plan of the Day

11:45 - 12:00	Arrive and Meet Teams
12:00 - 12:30	Admin, Demo and Safety Briefing
12:30 - 16:30	Build Time
16:30 - 17:30	Break for Food and “Tinker Time”
17:30 - 18:30	Challenge
18:30 - 18:45	Awards

(Times are subject to slippage since this is EARS and nothing ever goes quite right...)

Plan of the Labs



Safety

Scrap

The electronics scrap pile is a real scrap pile! Be aware that things may be sharp. We've done our best to discharge anything that may be charged, please be aware that there may be charged capacitors in the pile.

And, this goes without saying - **please don't plug anything into the mains!**

Sharp Objects and Inflatable Pools

The paddling pool is inflatable, which means its structural integrity is based on its ability to hold water. We **really** don't want to puncture the pool, since we don't want to have to explain to Mechanical Engineering Sciences why their labs have turned into an aquarium. Therefore, **no sharp objects are allowed in or near the pool** - any sharp patches on your contraptions must be covered (and Phil must approve).

LiPo (Lithium-Polymer) Batteries

Lithium-Polymer batteries are great when it comes to providing a good, stable power supply. However, they are also dangerous - please bear the following things in mind:

- **LiPo batteries will explode if punctured**! Please take extra care to handle the batteries with care and ensure that they are kept away from sharp materials.
- Do not directly connect the battery terminals together or short out the battery. Before we issue you with a LiPo battery you must demonstrate to us that there are no short circuits by powering your system from a bench supply.
- Do not charge the batteries yourself or otherwise connect them to a bench supply. Only authorised people (committee members and lab staff) may charge the batteries in the interest of safety
- Always use the battery with the provided fused battery harness. Under no circumstances should you cut the leads between the battery and the battery connector. (If you require more current than the fuse will allow, please talk to Phil who will provide you with a separate connector and fuse to build your own harness.)
- Try not to get the batteries wet. They should all be hermetically sealed and therefore waterproof, but it's best to not let them come into contact with water at all
- Don't heat the batteries up - this may cause damage.
- Never draw more than the maximum specified current from the batteries (20A) - it should be very hard to do this, especially because of the fuse, but if you blow your fuse it will take us a while to replace...

If something goes wrong:

- If you notice any swelling, outgassing or damage to the batteries please inform us **immediately**.
- If you are using the battery, disconnect at the power switch or by unplugging if it is safe to do so
- Step away from the battery

- **DO NOT** throw water on the battery. The cells contain lithium which reacts with water - so this will just make it worse!
- If the battery malfunctions in the paddling pool, do not make any attempt to save it or salvage your contraption.

WiFi

This challenge is using WiFi to communicate between the boats (running on ESP8266s) and the controlling device. We're providing some (2.4GHz) access points in the labs to allow you to test this as well as participate in the challenge. You may also wish to create your own mobile hotspots to test this - given that we anticipate a fair amount of network traffic. Also, **don't use EDUROAM** - Dan says it won't work.

The access points will have no internet connection but are used to connect both the ESP and the controlling device to the WiFi. The details of the access points are:

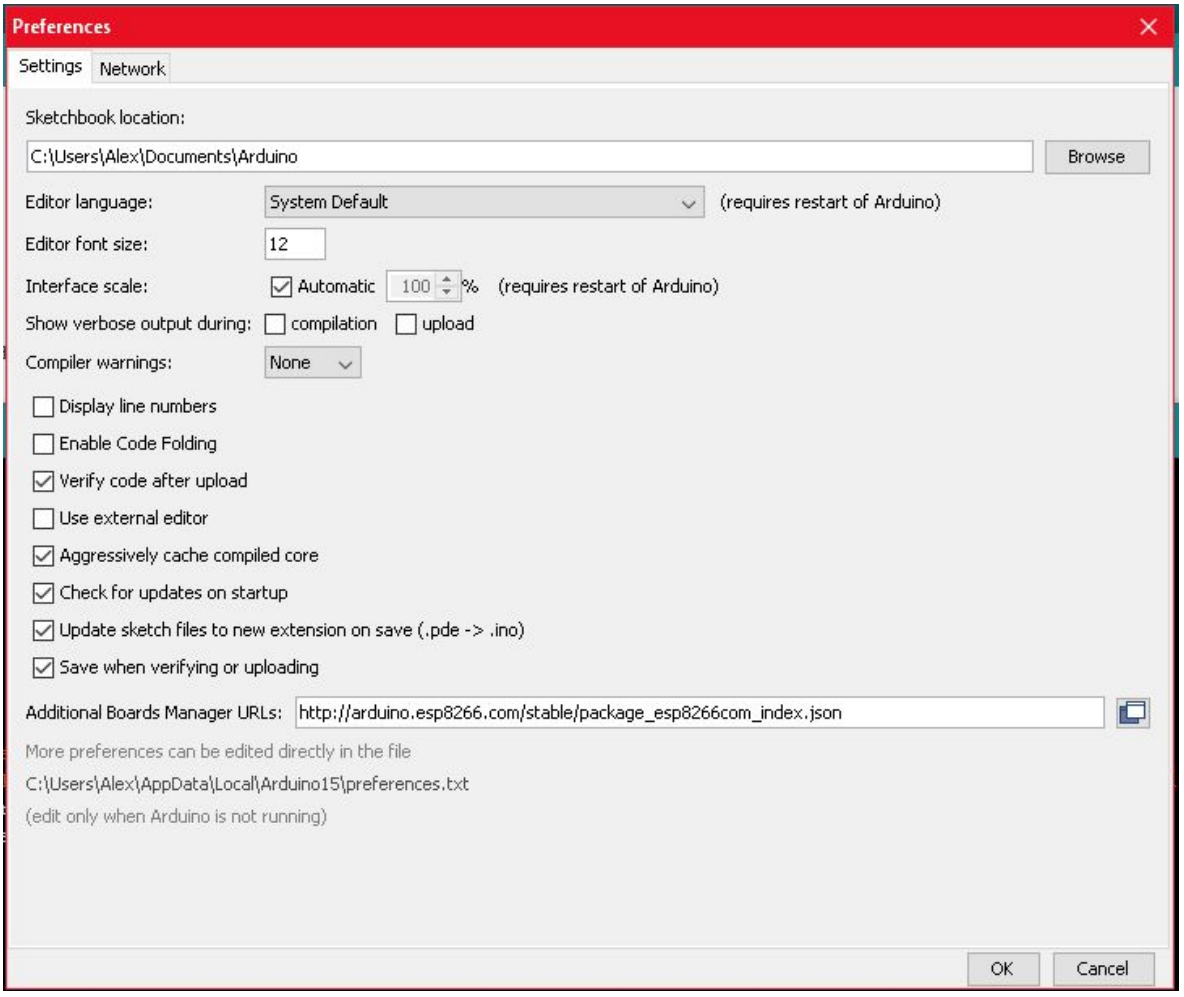
Location	**SSID**	**Password**
First Year Labs	EARSAP1	ears-wifi
Second Year Labs	EARSAP2	ears-wifi
Makerspace (hidden, but has internet (sort of))	makerspace	makerspace

Setup

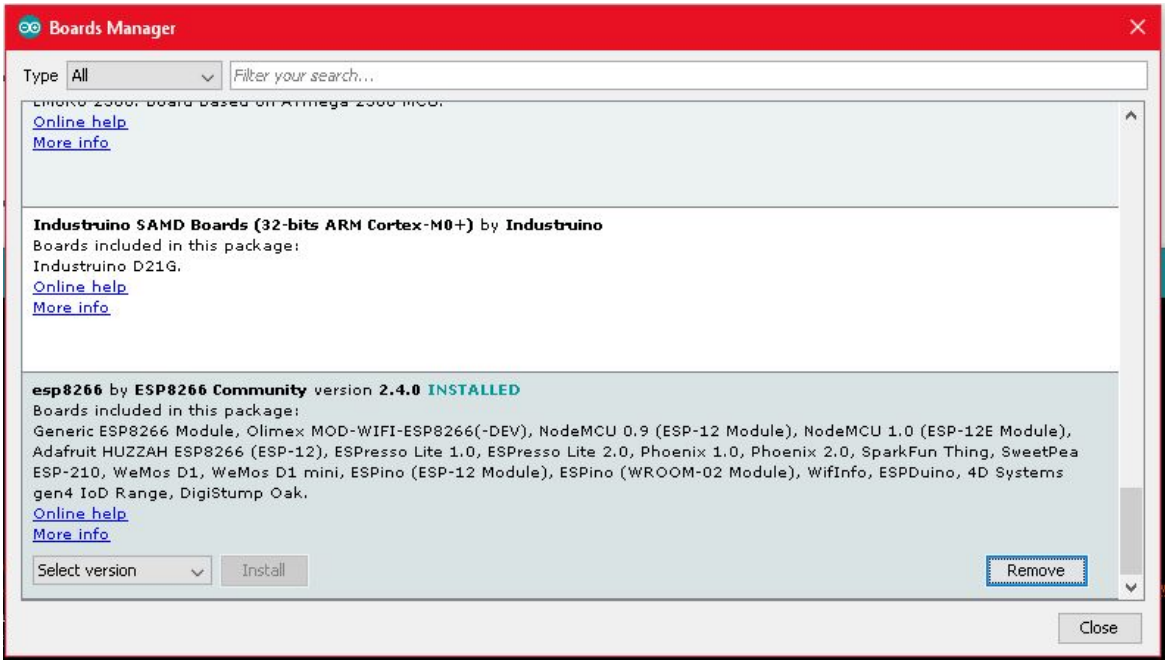
- Project available at https://github.com/SurreyEARS/scrapheep18
- Download library and client from here https://github.com/SurreyEARS/scrapheep18/archive/1.0.zip

ESP - Setting Up The Arduino IDE for ScraphEEp

1. (Recommended) Install the latest version of the Arduino Software from www.arduino.cc
2. Download the project from the link above, unzip it
3. In the Arduino IDE, navigate to **Sketch>Include Library>Add .ZIP Library** and select "lib-ScraphEEp-1K18" from the unzipped folder
4. Navigate to **File>Preferences** and set "Additional Boards Manager URLs" to **http://arduino.esp8266.com/stable/package_esp8266com_index.json** and click "OK"



5. Navigate to **Tools>Board>Boards Manager** and scroll down to find the entry for **esp8266 by ESP8266 Community**



6. Click on this entry and install the latest available version
7. To program the ESP correctly, you will need to tweak a few settings in the **Tools** menu. This will also make everything work the fastest:

a. Board: NODEMCU 1.0 (ESP-12E Module) - **this will make all the options below appear**
b. Flash Size - 4M (1M SPIFFS)
c. Debug Port - Disabled
d. Debug Level - None
e. IwIP Variant - v2 Prebuilt (MSS=536)
f. CPU Frequency 160MHz
g. Upload Speed - 921600
h. Port (select as needed when ESP is plugged in)
i. Programmer - AVR ISP

ESP - Writing Your Program

1. Add the ScraphEEp2k18 library to your code: **#include <scrapheep.h>**
2. (If you run into problems compiling this such as *" multiple libraries were found for "WiFiUdp.h" "* try adding **#include <ESP8266WiFi.h>** above this)
3. Above the line "void setup()", type "ESPControl control;" to create the ESP manager.
4. Once this is done, simply call "control.init();" inside setup() to connect your processor to the WiFi.
5. Finally, to start receiving data from the controller, put this code inside the loop() function:
   "uint8_t *data = control.processPacket();
   If (!data) return;"

6. All the data from the control software will be inside the data[] array, so simply access the data by calling data[the data ID].
7. The data IDs are as follows:
   a. MOTOR_A
   b. MOTOR_B
   c. C1
   d. C2
   e. C3
   f. C4
8. To save time doing this, you can work straight from Phil's example by clicking "File->Examples->ScraphEEp-2K18->PhilsRobot"

ESP - Interfacing and Use

The ESP and motor shield as given to you have:

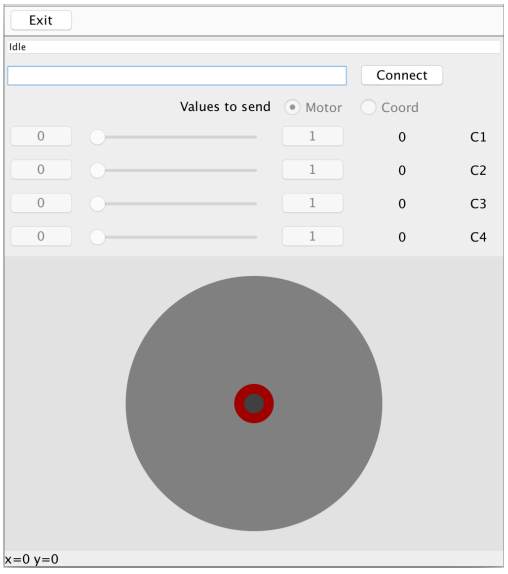
- Two separate power supplies (VIN for the ESP itself and VM for the motor driver output) on the blue screw terminals at the end -NOTE: **VIN<9V and VM<36V**
- Two two-terminal screw connections for motors on the blue screw terminals, labelled A and B.
- Nine Digital Pins labelled D0-D8 and set up to plug directly into a servo (see Technical Hints) - **NOTE: D1-D4 pins are shared with the motor driver chip, so can't be used as logic inputs/outputs unless the motor driver isn't in use**

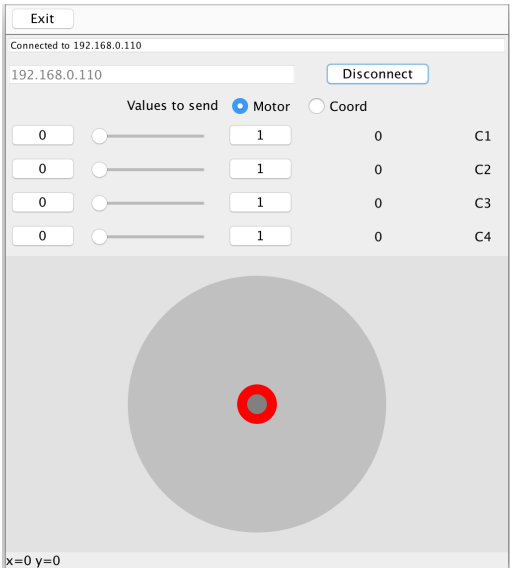
To Power the ESP and the motor shield you have a few options:

- **Single battery power** for both VIN and VM - the easiest way to power everything (as on Phil's demo). Place the pin bridge over the pins directly behind the blue screw terminals to connect VIN and VM together on the board
- **Single battery power with 5V regulated input** for VIN (as on Alex's demo). Find a 7805 5V regulator in the Makerspace and connect it heatsink-side-up in the three neighbouring screw terminals marked VM G VIN (you might also want to check a datasheet online). Feed your motor power to the VM and G terminals, and the ESP and servo power pins will be powered off 5V. This is really useful for stopping servo motors from blowing up if you want to use them
- **Dual Battery Power** for the motors and the ESP independently - connect the motor battery between VM and G and the ESP battery between VIN and G.

If using the LiPo battery, **please use the provided fused battery harness to connect the battery**. We will only be giving out batteries to teams who can demonstrate that their system is not short circuiting (to prevent any explosions etc) so before you get your battery it's best to power your systems from a bench power supply to test everything (the battery voltage will be 8.4V full, 7.4V minimum usable)

Client





1. Enter your ESP's IP address
   Written on a paper slip, or 192.168.0.1x where x is your team number *(if <10, add leading 0)*
2. Click *Connect*
3. 2 control types
   a. *Motor*: send differential motor values based on joystick location
   b. *Coord*: send joystick coordinates
4. 4 control sliders (*C1-4*)
   a. Customisable, sends a value between 0 and 255
   b. 0 and 1 buttons jump to the end of the slider
5. Joystick canvas
   a. For controlling motors
6. Click *Disconnect* to close connection, *Exit* to disconnect and close application

Technical Hints

Batteries

Provided your circuit doesn't have any shorts, we will supply you with a 2-cell Lithium-Polymer battery which provides a high current (enough to not be an issue) at 7.4V - this will increase to 8.4V for a fully charged battery.

You **don't have to** use this battery - for example, you might want to try and power a computer fan. These typically require a much higher voltage (ie. 18-24V) but a much lower current, which can be provided by stringing a few 9V batteries together in series. 9V battery harnesses are available in the Makerspace, and batteries are available from Alex (the Lab Technician - his office is next to the first-year labs and he **loves** being pestered).

However, don't forget that the ESP's input voltage (VIN) must not exceed 9V or things will start to go pop…

In this case, you might want to consider giving the ESP a separate supply or use a voltage regulator (the 7805 regulators in the Makerspace are rated for up to 35V input voltage)

Motor Selection

We have a lot of motors that can be used - there are some salvaged DC motors, some servo motors some really tiny useless little ones that Alex bought without thinking about it, and some geared DC motors with yellow gearboxes.

If you use the yellow geared motors, be aware that there are **THREE TYPES** with different speeds - we aren't really entirely sure why…

- White top, one pink shaft
- White top, two white shafts
- Black top, two white shafts

We advise that you check your motors before you glue them to anything (looking at you, Phil Bladen).

Motor Driver

The ESP motor shield that we've provided uses an **L293DD** H-Bridge motor driver chip, which can drive two motors. However, power-wise it's not the best, it's rated at 0.6A per channel with a peak non-repetitive current of 1.2A. From our tests, you can put a lot more than this through it for very short periods, but we don't recommend this at all. For the yellow geared motors, that'll probably be fine but if you want to drive a chunky motor (see Alex's example boat) then you might want a better driver chip.

We have some better ones (the catchily-named **ZXMHC3F381N8TC** single-channel H-bridge motor driver)

available as bare surface-mount packages - it's up to you to solder them to a breakout board (which we can also give you).

These can handle about 3.4A continuously, and apparently up to 20A in short pulses. We **really don't** recommend putting this much through them, since Alex managed to blow one up in testing.

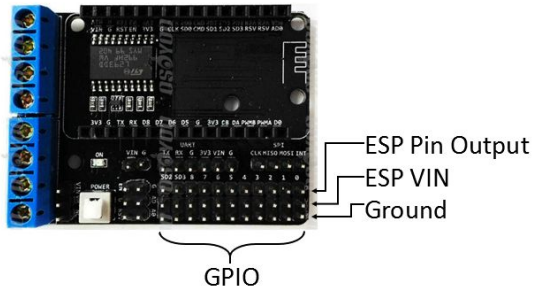
An example circuit is shown here using the driver chip - this is the simplest way to interface with the ESP using the output of the motor driver on the motor shield to drive it. It's up to you how to do this - we have level shifters available if you want to control it with 3.3V logic signals, and we recommend you check the IC's datasheet online before you use it.

My Motor Draws Too Much Current!

Please please please check how much current the motor needs with a bench supply before you try and use the motor driver! We don't have a lot of these chips. If you find that you're drawing too much current, the easiest (but not elegant) way to solve the problem is to reduce the voltage across the motor by putting some diodes on the positive supply line between the battery and the motor driver (this also works for the Vm input to the ESP motor shield). We have some chunky 1N5406 diodes in the makerspace, which can each handle 3A and will probably drop just over 1V when you pass a high current through them. Ask Alex if in doubt - he did this on his demo boat.

Driving Servo Motors

We have a small number of servo motors available for use. These can be used by connecting the servo directly to the ESP motor shield, which has conveniently located pins that allow us to do this. The pins provided match the pin ordering on the end of a servo motor lead.



Servo motors take their power directly from the VIN input (which also powers the ESP) and aren't really designed to run off the same voltage as our batteries. The easiest way around this is to provide a regulated 5V supply to the ESP - see what Alex did on his example.

To control the servos in software, the easiest way is to use the Arduino servo library which comes preinstalled in the Arduino IDE. We won't spoil the surprise and tell you how to do this, but if you google **"Arduino servo"** you'll find all the information you need.

Driving other stuff

The pins referenced in the "Driving Servo Motors" section can also be used as standard IO pins like an Arduino and can be controlled using digitalWrite and analogWrite - which produces PWM waveforms.

Apparently, PWM works on all of the ESP's pins and has a **maximum value of 1023** (as opposed to 255 on an Arduino). We haven't actually tested this thoroughly so let us know if you find otherwise…

Ducks, and how to collect them

All the ducks can be pushed on the water, but they also have a little magnet and a hook - just sayin' :)

Judging Criteria and the Challenge

Each team will have 5 minutes to attempt the challenge. There will be five ducks in the pool (because we have nine in total) and points will be awarded for every duck which is returned to the “duck vortex” beneath the large “Mother Duck”.

Teams who complete the challenge before the 5 minutes have elapsed will receive one bonus point for every fifteen seconds left over.

Other categories will be assessed by the judging criteria, outlined in the table below:

Category	**Judged / Based on**	**Points available (max.)**
Ducks collected	Number of ducks collected during the challenge	5
Time to spare	One point for every 15 seconds not used	20
Made Dan Happy	How happy the challenge attempt/contraption made Dan (as judged by Dan, based on his own secret scale)	5
Originality/Ingenuity	Approach to the problem and design of contraptions (ie. directly copying Alex or Phil scores 0 immediately)	5
Scrappiness	How much like a piece of electronics scrap the contraption looks (ie. Phil wouldn’t score a huge number of points - Alex probably wouldn’t either, but a floating oscilloscope would……)	5
Entertainment Value	How entertaining the attempt was to watch (does not include explosions - explosions are bad)	5
Did it float?	No points awarded if you sink, all awarded if you don’t. If you only sink a little bit - maybe we’ll go somewhere in-between.	5