EEERover Technical Guide

Department of Electrical and Electronic Engineering
Imperial College London

2017/8

Contents

Introduction

During the Spring Term you should put your high-level design ideas into practice by creating the circuits, software and hardware that will implement your EEERover. This guide has three main aims:

1. Document the additional hardware that you are provided with to get started
2. Specify the processes you will need to develop your design (e.g. ordering components)
3. Provide some implementation ideas that you may find useful

Most of the contents of this guide is advisory. You do not have to use your EEEBug or the items in the starter kit if you wish to pursue different ideas. The only obligatory content is in Section 2, which relates to the constraints you face for budget, facilities and ordering processes.

1 Starter Kit

You are given one starter kit per group — the contents are given in Table1. In addition to this, your EEEBug has been designed to support modification for work on this project. At the rear of the chassis is a mounting point for the Arduino and the central PCB has mounting points for the motor driver

Table 1: EEERover starter kit components

Qty.	Description
EEEBug add-ons	
1	TB6612FNG motor driver module
2	5 way PCB Receptacle, 2.54mm pitch
1	6 way PCB Receptacle, 2.54mm pitch
1	2×2 way PCB Receptacle, 2.54mm pitch
1	Arduino UNO WiFi
1	USB cable
4	M2.5x12mm machine screw
8	M2.5 nylon nut
Exoflask material simulator	
1	Exoflask
1	8mm button magnet

module. The breadboard, as you have already seen, is ideal for prototyping circuits but a number of mounting holes have also been provided for you to attach additional circuitry and components if you need more space. Finally, the design of the chassis is available to you in electronic format and you will have the opportunity to manufacture a modified version.

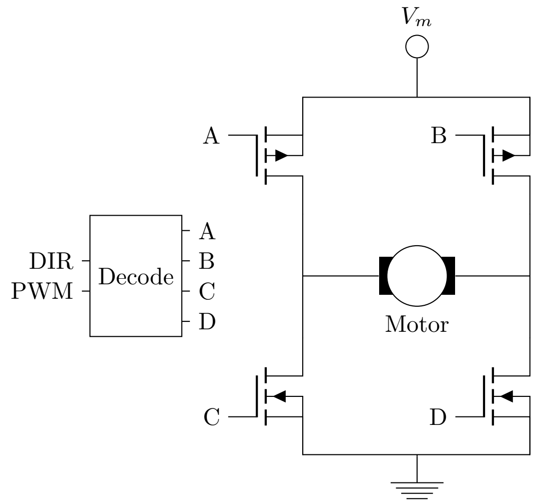
Motor Driver

The motor driver is a two-channel integrated *H Bridge* with decode logic, illustrated in Fig. 3. One channel controls the left motor and one channel controls the right motor. Each channel has two digital inputs, DIR and PWM, which are available on J2 of the EEEBug PCB (refer to the circuit diagram in the appendix). The digital inputs are decoded and used to control four *MOSFETs*, which are low-impedance switching transistors. The inputs can be connected directly to digital outputs on the Arduino. DIR controls the motor direction and the motor speed can be controlled by applying a rectangular waveform to the PWM input and varying the duty cycle. The motor driver plugs into J4 and J5 of the EEEBug PCB.

Arduino

The Arduino UNO Wifi is a embedded computing platform with internet connectivity based on the Arduino form factor, libraries and development tools (see Fig. 2). It is programmed in C++ and features digital inputs and outputs, inputs with digital-to-analogue conversion and PWM outputs.

You are provided with demonstration code (see the labweb website) that shows the use of basic functions of the Arduino. The WiFi interface is used to host a basic webpage, which you will see hard-coded into the demonstration code. The webpage provides two links which, when pressed, turn on and off the onboard LED. Use the Arduino Serial Monitor to connect to the board via USB and you will see some debug messages reporting on the connection



DIR	PWM	A	B	C	D	Motor
x	0	V_m	V_m	V_m	V_m	off
0	1	0V	V_m	0V	V_m	forwards
1	1	V_m	0V	V_m	0V	reverse

Figure 1: A single channel of the H Bridge motor driver with the truth table for decoding logic inputs DIR and PWM

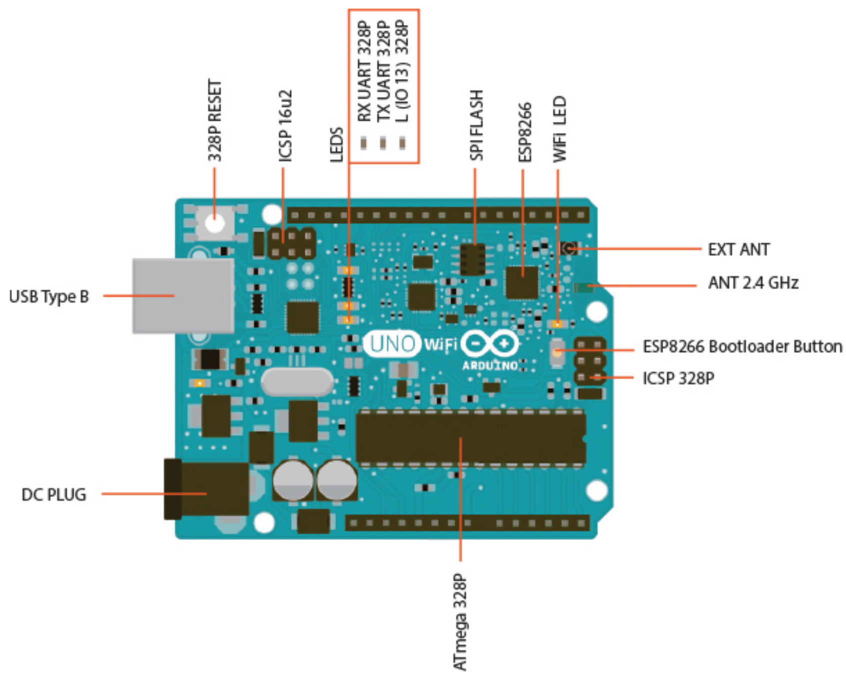


Figure 2: Diagram showing the key components of the Arduino Uno Wifi module. From `https://www.arduino.cc/en/Guide/ArduinoUnoWiFi`

status. Whenever the webpage is retrieved you will see the HTML request made by the web browser.

Arduino is a popular platform for prototyping and you will find a lot of information and many code examples on the internet. Use these resources but beware that many are provided by third parties and may not be well-written or even correct! The WiFi communication functions are provided by the WiFi Link library, which is an official library but is not installed by default. To install it in the Arduino IDE go to `Sketch` → `Include Library` → `Manage Libraries` and search for it. WiFi Link inherits its API from the WiFi library, which is documented here: `https://www.arduino.cc/en/Reference/WiFi`

WiFi Link comes with its own set of example sketches. Once you have installed the library look in `File` → `Examples` → `WiFi Link`. The demonstration code from Labweb is based on the `SimpleWebServerWiFi` example.

WiFi Connection

The demo code is configured to connect to a stand-alone WiFi network `EEERover` that is set up in the lab. Before uploading the demonstration code, change the IP address defined in the code to the unique IP address written on the back of your Arduino board. To reach the webpage, connect to the EEERover network with a laptop or other device (password `exhibition`) and use a web browser to access your IP address, for example `http://192.168.0.1/`.

You will see in the code how you can define which WiFi network the Arduino will connect to. However, it is not possible to connect to networks secured with WPA2-enterprise, including the Imperial College network.

The Arduino can also host its own WiFi network. Connect to an SSID of the form `Arduino-Uno-WiFi-xxxxxx` and go to `http://192.168.240.1/`. Make a note of your Arduino's unique SSID so that you can identify it if there are others present. Please connect via the EEERover network if you are in the lab because communication over many different WiFi networks will cause interference and unreliable connections.

Power

The Arduino can be powered by USB or by the EEEBug batteries. There is also a power jack but that is not needed. To connect to the EEEBug batteries you will need to make the following connections between the Arduino and the EEEBug PCB:

- 5V→5V
- GND→GND

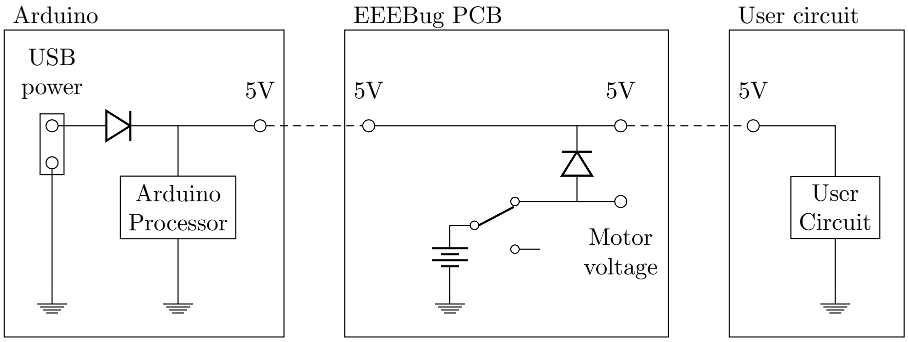


Figure 3: A simplified diagram showing the connection of the Arduino power supply, battery and motor supply

Always check these connections before switching on the power. You may damage the Arduino if you make a mistake!

The Arduino will automatically take power from the USB cable if battery power is not present. The 5V terminals on the Arduino are powered by USB in this case so the USB connection will continue to power other circuitry connected to 5V if the battery is turned off. However, recall that the EEEBug PCB contains a diode between the battery and the 5V terminals. The diode prevents the 5V power rail from feeding back to the motor driver, so the motors will only operate if batteries are inserted and the EEEBug power switch is turned on.

Mounting

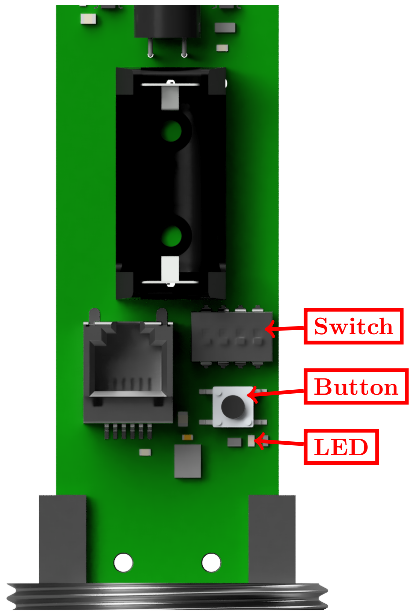
The Arduino can be mounted on the space at the back of your EEEBug. Insert the M2.5x12mm screws from the underside of the chassis and secure them with one nylon nut per screw. The Arduino is then located on top so that the nuts act as a spacers to hold the Arduino clear of the chassis. The additional nuts can be added on top of the Arduino to secure it but not all of these will fit due to the proximity of components to the mounting holes.

Exoflask material simulator

Your prototype will be developed and tested using Exoflask material simulators: electronic devices which can replicate the properties of the materials that you are searching for. An Exoflask is provided to you to aid your development. Fig. 4a shows an Exoflask, while Fig. 4b shows the features of the internal circuit board used for configuration. Unscrew the bottom end cap of the exoflask to access the circuit board.

The Exoflask is configured by setting a group of DIP switches on the PCB. A push button is used to turn it on and off, and an LED indicates when it is active. A flow chart depicting the operation of the Exoflask is

(a) Exoflask appearance



(b) Exoflask configuration

Figure 4: Exoflask Images

given in Fig. 5. From standby, press the button to turn on and observe the LED to determine whether the battery has sufficient charge. After the battery display sequence is complete the Exoflask begins emitting signals according to the mode set up on the DIP switches. The LED flashes a code at intervals to confirm the mode selection and indicate operation.

Pushing the button during operation will reset the Exoflask — this is necessary to change the operation mode. A second press, occurring during the battery sequence, will return to standby mode. Power consumption during standby is very low and the battery does not need to be removed. If the Exoflask is left operating for five minutes it will automatically return to standby mode to preserve the battery.

The Exoflask cannot simulate the magnetic property in an readily-configurable manner so you are provided with a small magnet to test this functionality. A realistic magnetic field density will be achieved at a distance of approximately 5mm from the magnet face.

The emitted acoustic and infrared signals are strongest at the top end of the Exoflask. The permanent magnet will also be strongest at the top with a vertically-orientated flux. However, the direction of the flux (polarity of the magnet) is unknown. The radio signal is not precisely located, although you can observe the orientation of the antenna coil on the internal PCB.

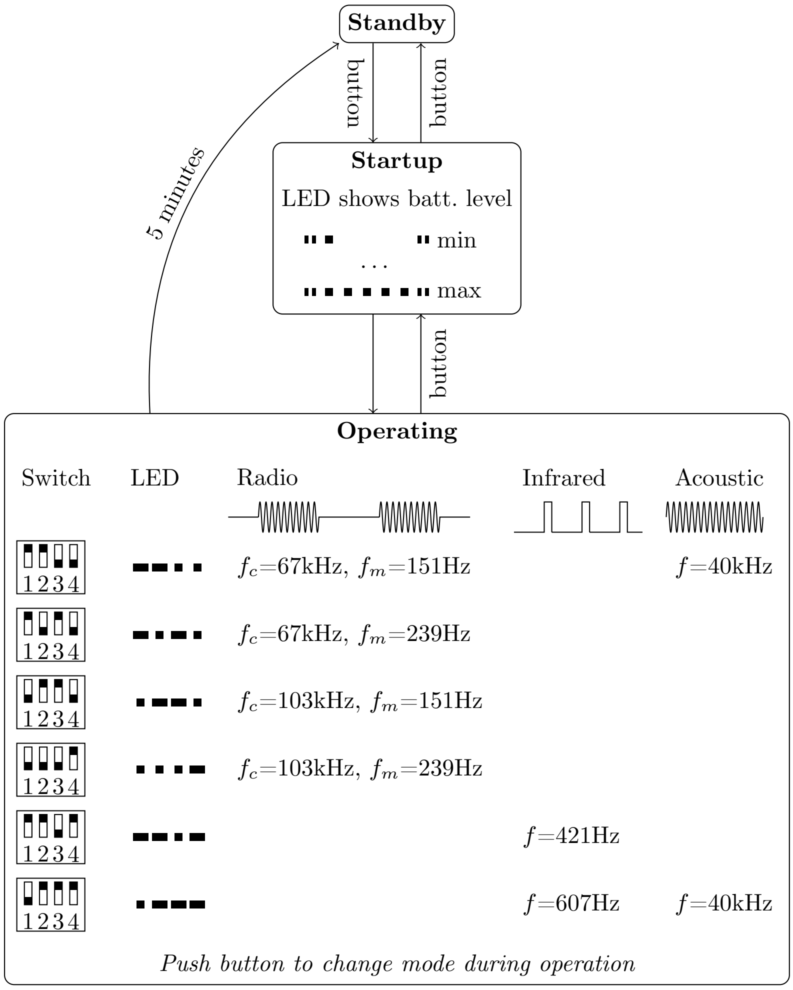


Figure 5: Exoflask operating flow

2 Components and Construction

Ordering components

A tool for ordering components is provided in the downloads section on Labweb: `https://intranet.ee.ic.ac.uk/intranet/labweb/downloads/`. Each project group has a budget of £50 to cover all the additional components you'll need for designing and building your prototype. When you first open the ordering system you will need to allocate a treasurer for the group, who will be responsible for managing the budget and placing orders. Each order you create must be approved by another group member. Group members receive emails when an order is placed and the approver will receive an email when the parts are available for collection from Stores on level 1, near the lifts. Orders, assuming the items are in stock, will take up to a week to arrive. You can order some components directly from stores stock and these will usually be available for collection sooner.

The ordering tool contains links and a browsing window for choosing parts from the department's approved suppliers. When you have selected a part, copy the information into the form to add it to the order. Please be aware of the following rules and common pitfalls when ordering components:

- You must only order items that you can use safely. If you have any doubts about this then talk to a member of staff before ordering. Do not order chemicals (including glue), tools or mains-powered equipment/parts without consulting staff.
- Always check that the supplier holds sufficient stock. If an item is out of stock it will be placed on back-order and it may take months to arrive. Try to find an alternative part or supplier instead.
- Some suppliers offer stock from sister companies around the world. Do not order any items which warn about additional delivery charges.
- Many electronic components are *surface mount*. This means that they sit on flat pads on a PCB instead of having flexible leads to insert into a prototyping board. They are also usually very small. You will find surface mount components difficult to use so avoid them by checking item descriptions carefully.
- You need to maintain the goodwill of the stores staff to order parts for this project and future projects. Do not chase the staff for your order until one week has passed from placing the order. It is your responsibility to check parts for suitability and availability.

Ordering a new chassis

You can use the ordering tool to buy a new laser-cut chassis for your EEERover. Since these are made to order there are two limited windows for purchasing them:

1. Order by 17:00 on 2 March for delivery on 12 March
2. Order by 17:00 on 27 April for delivery on 7 May

You can download a design file for the EEEBug chassis from labweb in PDF format — it is recommended that you use this as a template. You will see that cut lines are drawn in red and engrave lines are drawn in blue. Laser cutting removes a tapering width of material known as a *kerf*, which for 3mm acrylic is typically 0.1mm at the bottom of the material and 0.3–0.4mm at the top. Hence, cut edges are not vertical, external dimensions will be slightly smaller than drawn and internal dimensions slightly larger than drawn.

Designs for laser cutting must be submitted in PDF format. Two sizes are available to order and the entire design must fit within the quoted dimensions. Multiple separate parts can be ordered as long as they are laid out in an area that fits the selected order size. Each part must be engraved with your group name. You are strongly recommended to check your design by printing it on paper first — the scaling will be identical as long as the same PDF file is used.

3D printing and Workshop facilities

A basic workshop is available in the lab. To use it talk to Vic and be prepared to explain exactly what you want to do and demonstrate that you are capable of doing it safely. You can also have components 3D printed if you are willing to learn the necessary design techniques independently. Talk to Amine in the technicians' office if you would like to use 3D printing.

3 Implementation Hints

The table of material types from the design brief as follows:

Name	Property 1	Property 2
Gaborite	67kHz radio modulated at 151Hz	Acoustic signal at 40.0kHz
Nucinkisite	67kHz radio modulated at 239Hz	Magnetic field
Durranium	103kHz radio modulated at 151Hz	Magnetic field
Brookesite	103kHz radio modulated at 239Hz	None
Cheungtium	Infrared pulses at 421Hz	None
Yeatmanine	Infrared pulses at 607Hz	Acoustic signal at 40.0kHz

Radio

Four material types emit radio waves, but you can choose the amount of discrimination that you can provide. Is it sufficient just to detect any radio frequency? Do you need to distinguish between the carrier frequencies? Do you need to measure the modulation frequency?

It is recommended that you use a tuned coil antenna to pick up the signal. This is essentially an air-cored inductor with a reasonably large diameter, which you can make by coiling wire. This inductor can then be used in a tuned circuit that will be most sensitive at the resonant frequency. There are only two carrier frequencies to detect so you do not need the continuous adjustment of tuning that you get with an audio radio receiver, for example.

The signal amplitude generated by your antenna will be small so it will need amplification before you can use it to indicate detection. The Arduino microcontroller will struggle to sample the radio signal at its carrier frequency so you will need to perform *envelope detection* to produce a signal which indicates the amplitude of the signal. The easiest way to do this is by rectification. The modulation is a form of *Amplitude Modulation*, where the amplitude of the signal varies between 1 and 0 at f_m, the modulation frequency. f_m can be determined using an analogue circuit such as an frequency-to-voltage converter, or you could count the signal period with the microcontroller.

Infrared

You have already made a light sensor as part of the EEEBug and most silicon-based photosensors (such as the EEEBug phototransistor) are just as sensitive to infrared as visible light. However, the optical power given by the Exoflask is weaker than the light source you used for the EEEBug so a greater amount of amplification may be needed than the bipolar transistor which you used before. You may need to measure the pulse frequency of the signal and, as with the radio modulation, this can be done with analogue or digital methods.

Ultrasonic and magnetic

The magnetic field of the Exoflask is static, meaning that it will not induce a current in a coil of wire unless that coil is moving. A moving coil is a possibility, but there are also sensors and switches that can detect a static magnetic field. Test any new components and read their datasheets to determine if any amplification or signal processing is required.

Sensors are also readily available for detecting ultrasonic signals. Amplification may or may not be necessary, depending on the sensitivity of your sensor. As with the radio receiver, rectification will convert the high-frequency audio signal into a voltage representing the amplitude. Most ultrasonic transducers have a resonant frequency at which they are most sensitive. Check component datasheets to confirm the frequency sensitivity of any sensor you are planning to use.

Opamps

You may wish to implement amplification or filtering using an opamp. There are many different opamps and they have different specifications. Some manufacturers have parametric search facilities on their websites to narrow down the options based on your specifications. Look at the appendix of the opamps experiment instruction to see an example of how the LT1366 device was selected for that application.