# ALBIR PIXYBOT RACE

Contents

[ALBIR PIXYBOT RACE 1](#_Toc64365782)

[Module repository 2](#_Toc64365783)

[Assembling your pixyBot 2](#_Toc64365784)

[Hardware usage notes 2](#_Toc64365785)

[Setting up your software environment 3](#_Toc64365786)

[Flashing an OS image to your SD card 3](#_Toc64365787)

[Connecting your Pi to your network 3](#_Toc64365788)

[Connecting to your Pi using SSHFS 3](#_Toc64365789)

[Running exercises 5](#_Toc64365790)

[Editing ALBIR scripts 5](#_Toc64365791)

[Running ALBIR exercises 5](#_Toc64365792)

[Setting up your pixycam 5](#_Toc64365793)

[Submitting 5](#_Toc64365794)

[Common errors and fixes 6](#_Toc64365795)

[My Pi won’t boot 6](#_Toc64365796)

[I get a segmentation fault when I run a script 6](#_Toc64365797)

[My Pi crashes/reboots sometimes when I drive the servos too hard 6](#_Toc64365798)

# Module repository

* <https://github.com/dragonflyneuro/ICL-BioEng-ALBiR-PixyBot>

# Assembling your pixyBot

* Watch <https://www.youtube.com/watch?v=y1_FM6MfdNg>.

# Hardware usage notes

The robot assembly is covered in the assembling instruction video. Here we will cover some principles for the use of hardware:

* To minimize the weight and size of the robot, the onboard battery can only allow the robot to operate freely for up to 1-1.5hr during normal use. The battery hat has 5 leds to indicate the battery charge. Stop running the robot for charging when the battery reaches 20% level (or only one led on).
* While programming the robot, you may connect USB power to the battery hat to charge the battery and power the raspberry pi. Unplug the USB power only when you are ready to run the robot freely. You will have to shut down the Pi before unplugging.
* The current robot design requires a flat surface to run. Avoid tabletop if you worry about the robot running off the edge (or just be very mindful). Smooth floor is the best.
* While the Pi has a micro-HDMI port, we encourage you to access the operation system directly via WiFi. See instructions in this guide on how to set it up. The remote access and uploading/downloading files require very little data so connecting via a mobile phone hotspot should be sufficient.
* You could also work on the python code on your computer and upload to the robot via remote access.
* **PLEASE MAKE SURE TO RUN THE COMMAND sudo shutdown now or sudo reboot now to shutdown the Pi/reboot the Pi safely. Turning the battery HAT off, pulling the power cord or plugging/unplugging any USB or servos without shutting down properly may corrupt your SD card.**

# Setting up your software environment

## Flashing an OS image to your SD card

1. Only follow these steps if your SD card is not flashed already. You can tell if it is already flashed if you see a “boot” partition when you plug your SD card into your computer.
2. Download the ready-to-go Pi operating system image on the module Team > General > Files > Class Materials
3. (WINDOWS) Use Win32 Disk Imager (<https://sourceforge.net/projects/win32diskimager/>) to flash the SD card. Select the SD card drive letter on the right and use the file picker to choose your image file. Click “write” to flash the SD card.
4. (MAC) Follow this link (<https://www.raspberrypi.org/documentation/installation/installing-images/mac.md>) to flash the SD card.

## Connecting your Pi to your network

1. Insert the SD card into your computer, your computer should recognise a boot partition and several others (YOUR OPERATING SYSTEM MAY TELL YOU TO FORMAT THE DISK IN DRIVE X BEFORE YOU CAN USE IT. CANCEL THIS PROMPT). Go into the boot partition and create a file called “wpa\_supplicant.conf” (an example file is on the module repo. Open this file in a text editor and fill it with (**the Pi cannot see 5Ghz networks**):

ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev

update\_config=1

country=<Insert 2 letter ISO 3166-1 country code here>

network={

ssid="<Name of your wireless LAN>"

psk="<Password for your wireless LAN>"

}

Your country code should be “GB” if you are in the UK. If not, please refer to <https://en.wikipedia.org/wiki/ISO_3166-1>. Fill in your wifi SSID and psk (password) and save the file.

1. Boot the Pi by turning on the battery HAT or plugging in a microUSB cable to the battery HAT. The green LED on board the Pi should flash.
2. You now must find the Pi’s IP address. We recommend using Fing (Android: <https://play.google.com/store/apps/details?id=com.overlook.android.fing&hl=en_GB&gl=US>, iOS: <https://apps.apple.com/us/app/fing-network-scanner/id430921107>) on your smartphone to scan your network and find the device with hostname **raspberrypi**. The smartphone must be connected to the same network you connect the Pi to.

## Connecting to your Pi using SSHFS

1. Now we need a SSHFS client to connect to the Pi. This client will make it easy to edit files on your PC then send files to the Pi and run scripts.
2. (WINDOWS) We recommend MobaXTerm (<https://mobaxterm.mobatek.net/>).
3. (MAC) Refer to <https://www.raspberrypi.org/documentation/remote-access/ssh/sshfs.md> and <https://github.com/osxfuse/osxfuse/wiki/SSHFS> to get SSHFS working on MacOS.
4. Connect to the Pi using the SSHFS client and IP address you found. **The default username on the Pi is pi and the default password is raspberry.**
5. Now you should be greeted with a command line interface which you can use to navigate the Pi and run scripts!

# Running exercises

## Moving files to your pixyBot

* Download files to your computer, then move them into your pi by navigating to Documents/ALBIR using the command cd Documents/ALBIR then dragging and dropping. This folder already has some library files you need to run the pixyCam.

## Editing ALBIR scripts

1. You can either edit files on the Pi using nano <File name>.py or edit them on your computer by downloading the class files from Blackboard Learn or downloading them from the Pi using your SSHFS client.
2. If you are using your computer to edit, you can upload the files to the Pi using the SSHFS client.

## Running ALBIR exercises

* Navigate to Documents/ALBIR using the command cd Documents/ALBIR
* Run scripts using sudo python3 <File name>.py
* Prematurely end scripts using ctrl+c
* If it is a .m file run the script using MATLAB on your computer

## Setting up your pixycam

1. Install “PixyMon v2” on your computer (<https://pixycam.com/downloads-pixy2/>)
2. You can connect the pixyCam to your computer using a microUSB cable. **Please make sure to shut down the Pi when plugging/unplugging any USB peripherals OR the servos, as they are not designed to be hot-swappable and doing so may damage the Pi.**
3. Open PixyMon v2 and you should see a small stream of what the pixyCam sees.
4. Follow this guide to tune your PixyCam: <https://docs.pixycam.com/wiki/doku.php?id=wiki%3Av2%3Asome_tips_on_generating_color_signatures_2>

## Submitting

1. Submit your code files/videos using Blackboard Learn.
2. Submit your robot by bringing it into South Kensington campus. Details will be announced soon.

# Common errors and fixes

## My Pi won’t boot

1. Turn off the switch on the battery Pi Hat, then try plug in the charging USB cable in for a second then unplug. Turn the battery Pi Hat on.
2. Check if you have connected the servo PCB correctly.
3. Contact Daniel Ko on teams ([dsk13@ic.ac.uk](mailto:dsk13@ic.ac.uk)) if these steps do not fix the issue. You may have to get a replacement part.

## I get a segmentation fault when I run a script

1. Make sure you are using sudo
2. Make sure the Pi and the Pixycam is securely connected using the USB cable.
3. Contact Daniel Ko on teams ([dsk13@ic.ac.uk](mailto:dsk13@ic.ac.uk)) if these steps do not fix the issue. You may have to send him the scripts you were using.

## My Pi crashes/reboots sometimes when I drive the servos too hard

1. Charge the battery. Try keep above 1/4 charge.
2. Deactivate the HDMI port to try save some current bandwidth. You can do this by editing /etc/rc.local (with sudo nano /etc/rc.local) by adding the following lines above exit 0. Remember to reboot after:
   1. #disable HDMI
   2. /usr/bin/tvservice -o
3. Change the last argument to any calls to setMotorSpeeds to 1 i.e. setMotorSpeeds(X, Y, Z, 1). This turns the smooth speed transition mode on. Read the documentation in pixyBot.py to find out more.
4. Contact Daniel Ko on teams ([dsk13@ic.ac.uk](mailto:dsk13@ic.ac.uk)) if these steps do not fix the issue. You may have to send him the scripts you were using or get a replacement part.