Leeds Trinity University

A warm welcome from

The School of Computer Science at Leeds Trinity University

WHAT'S COVERED…

1. Types of Jobs in Computer Science
2. Our Computer Science Team
3. Course Structure including Placement
4. Why LTU for Com Science?
5. Basic Python, and Programming Activity

COMPUTER SCIENCE TEAM


Professor Yanguo Jing


Dr Nicole Danino

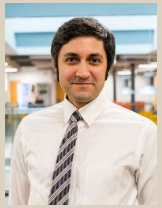

Dr Antesar Shabut


Mr Jim Diockou


Dr Aliyu Lawal Aliyu


Dr Lesley May


Dr Xin Lu


Dr Yashar Baradaran Shokouhi


Dr Nick Mitchell

Examples of research within the school:

- AI in healthcare and Biomedical Signal Processing
- Cyber Security and the future of air commute
- Software Defined Networks (SDN) & 5G
- Opensource research
- Globaliser Mobile App

POTENTIAL JOBS IN COMPUTER SCIENCE

Web Designers, App Developers, Games Designers, Cyber Security, Algorithm Developers, Data Analytics, AI & Machine Learning Lead, …

Read more …

Available Courses

Currently available programme:

- BSc Computer Science (BCS Accredited)
- BSc Computer Science with Artificial Intelligence
- BSc Computer Science with Game Development
- BSc Computer Science with Cyber Security

- MSc Data Science and AI (Scholarships available)
- MRes and PhD Research

Foundation year available.

COURSE STRUCTURE

BSc Computer Science

bcs Accredited Degree

How we structure our course:

Year 3

Varies by chosen course, modules include topics such as:

- Secure development
- Artificial intelligence
- Game development
- Responsible computing
- Data science
- Individual project

Year 2

Varies by chosen course, modules include topics such as:

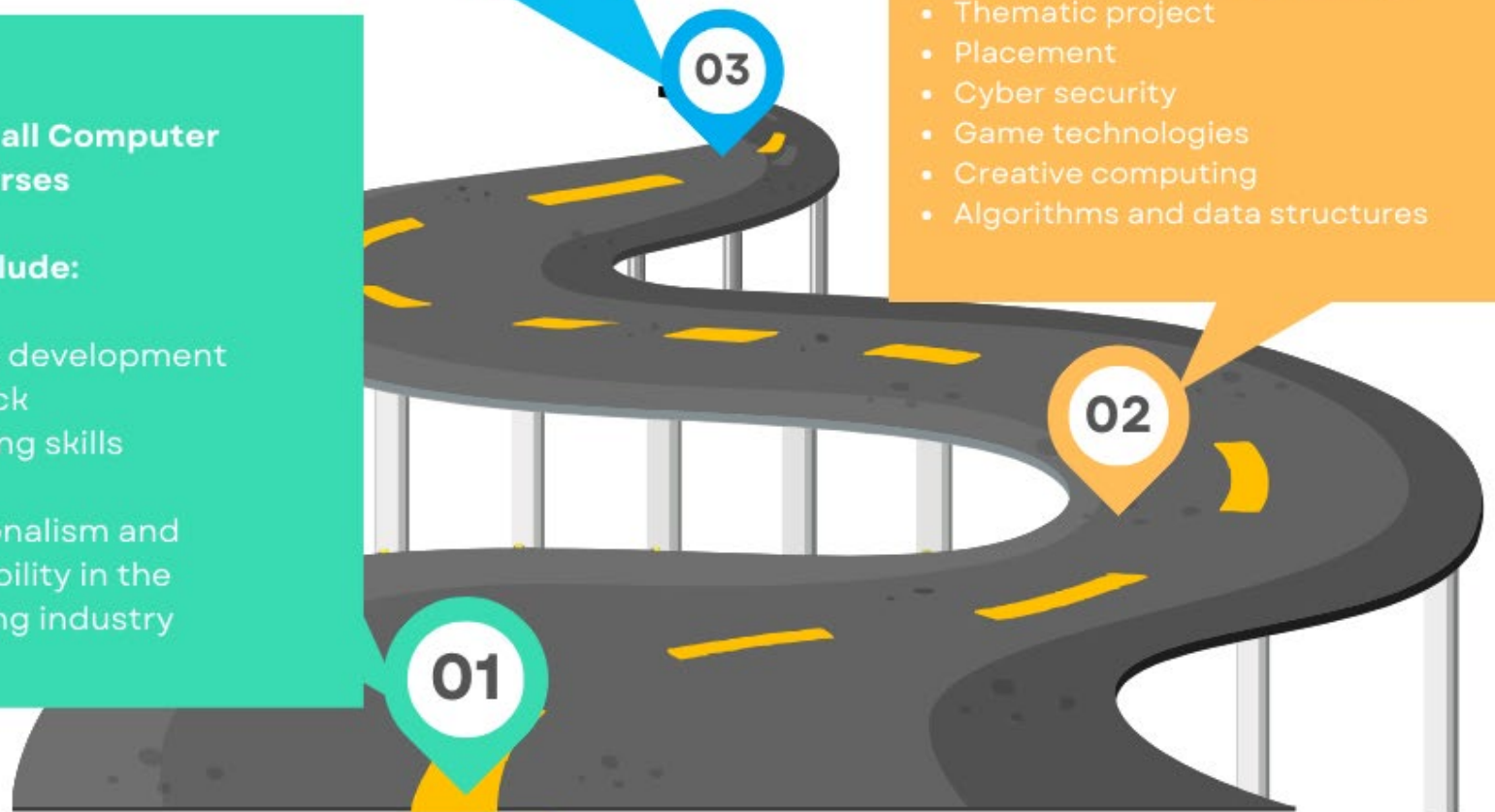
- More advanced programming
- Thematic project
- Placement
- Cyber security
- Game technologies
- Creative computing
- Algorithms and data structures

Year 1

Common to all Computer Science courses

Modules include:

- Software development
- Tech stack
- Computing skills
- UX
- Professionalism and employability in the computing industry



Foundation Year

Most of our undergraduate degrees can be taken as a four-year course, beginning with the associated Foundation Year for your chosen degree.

- Aims to develop your academic knowledge
- Introduces you to key concepts in computing
- Develop technical skills in preparation for the BSc course
- Has lower entry requirements than our standard
- three-year degree

WHY LTU?

Most of our graduates are in employment or further study 15 months after graduating (>93%)

- A supportive university community
- Staff availability
- Small group teaching
- Industry-led teaching and mentoring

WHY LTU?

- Industry & Research Led Teaching
- Visiting Lecturers from
  - Industry
  - Royal Academy of Engineering
- Industry visits, events and placement opportunities

Professional work placements

Partners:
- UST
- Apporto
- Hippo Digital

Example of other companies:
- Legal Tech
- Farnell
- Samsung
- And many others

Leeds Tech City

Examples of the range of tech companies actively recruiting in Leeds and West Yorkshire include:

Media
Health
Fashion
Gaming
Enterprise Software
Marketing
Fintech
Wellness Beauty
Kids
Food

Event Tech
Transportation
Energy
Home living space
Service Provider
Sports
Real Estate
Job Recruitment
Space
Travel

Finance & Banking
Property Sector
Legal Firms
Government
Universities - Academic route
Energy and Renewable Energy
Net Zero Carbon Sustainability
Construction
Big Tech Companies
Automation and self-driving vehicles
Cyber Security Analysts

Study abroad

We partner with universities, institutions, employers and charities across the world to give you amazing opportunities to have an international experience.

This may include:

- Studying abroad for a full academic year
- International volunteering
- Short-term international trips

Leeds Trinity University

Contacts

Dr Yashar Baradaranshokouhi

Yash.b@leedstrinity.ac.uk

Dr Nicky Danino (Head of School)

n.danino@leedstrinity.ac.uk

Thank you for listening

Any questions?

Leeds Trinity University

Part 2: Activities

Task 1:

Simulation

Task 2 & 3:

Hands-on Experience with Python and Raspberry Pi

Task 4:

Pitch your app

Leeds Trinity University

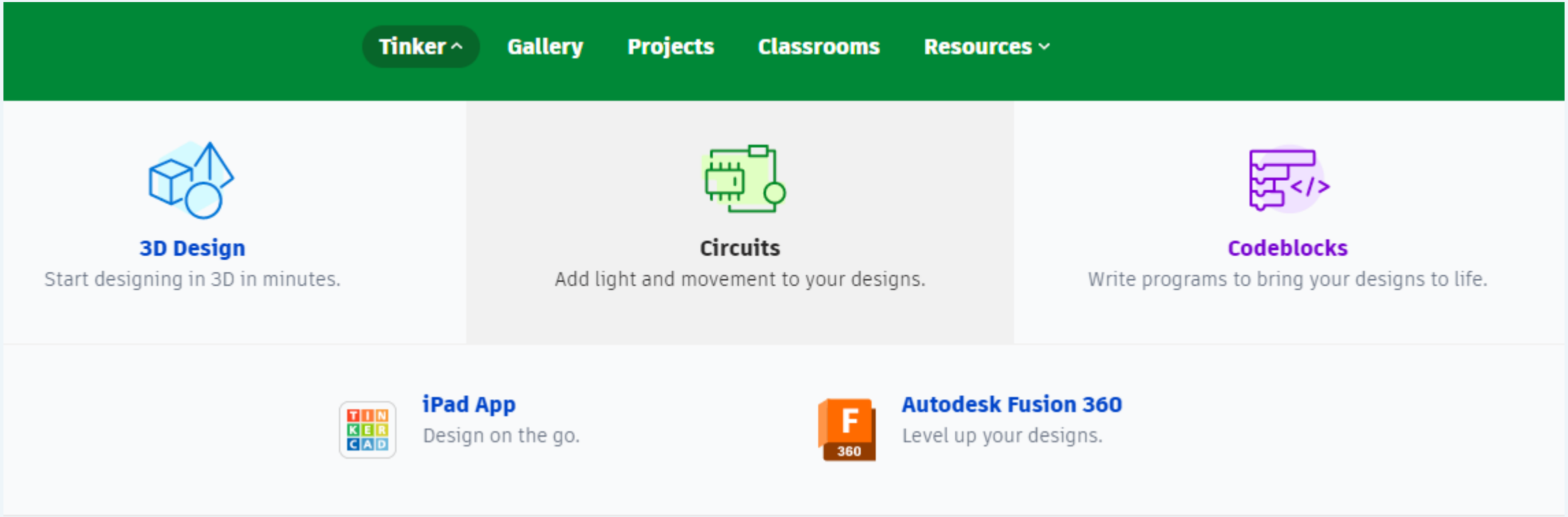
Task 1: Simulation

Turn on an LED with an Arduino board

Platform: TinkerCAD

Arduino Board & TinkerCAD

Navigate to Circuits | Tinkercad



Blink an LED with Digital Pin

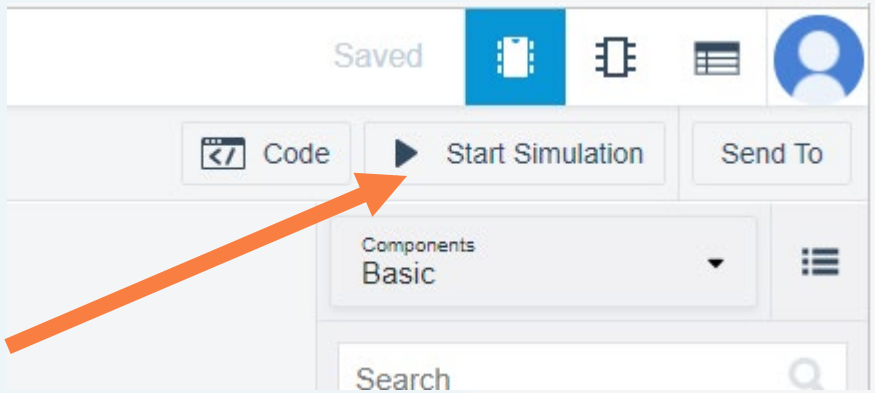
Follow the link to access the laboratory

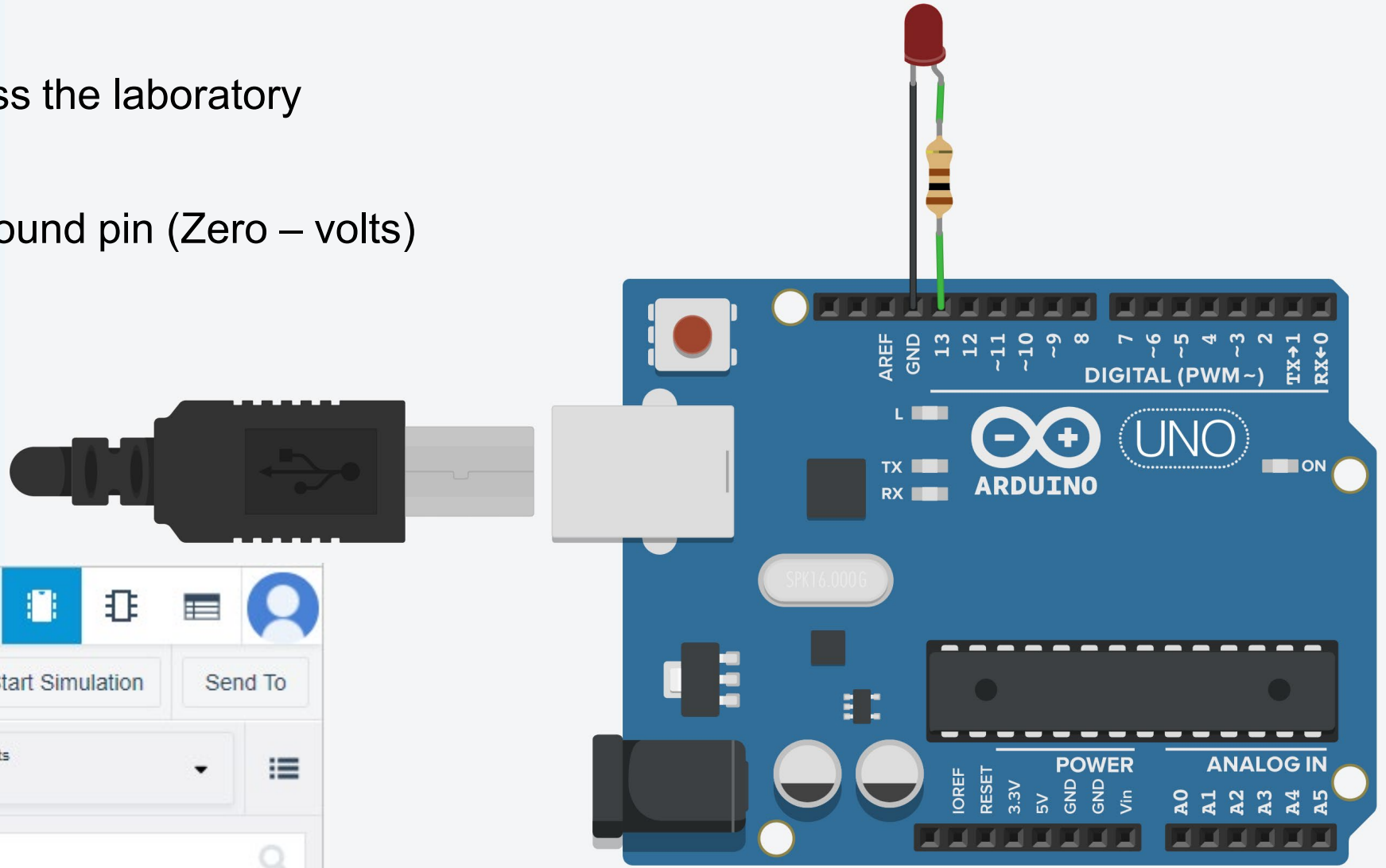
GND stands for the Ground pin (Zero – volts)

Pin13 is a digital Pin

Pin13 is set to 5 volts.

Run the simulation





Leeds Trinity University

Task 2:

Let’s try it with Raspberry Pi

Leeds Trinity University

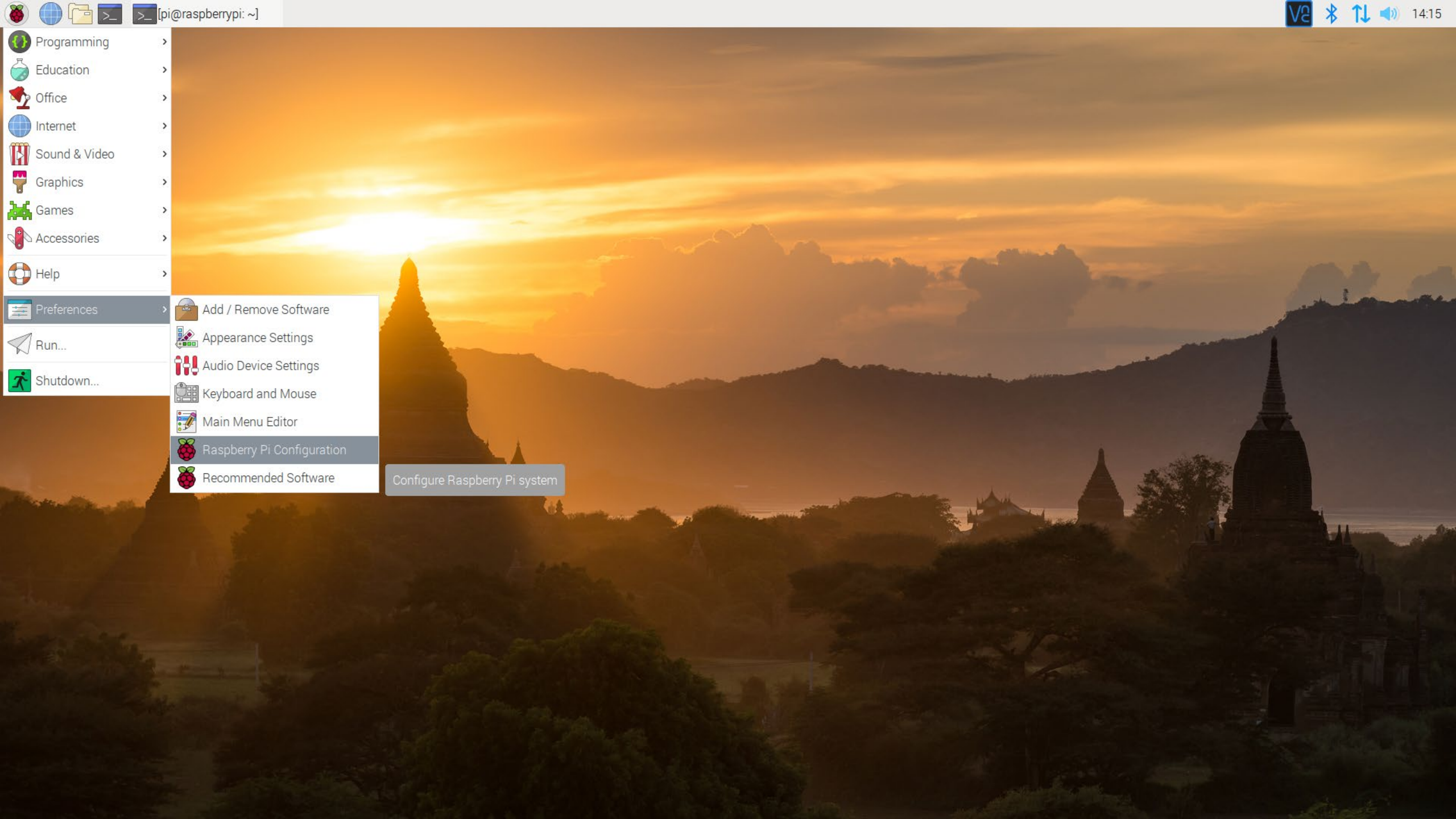
Setting up the Raspberry Pi

1. Before connecting the power to your Raspberry Pi
2. Connect the keyboard, mouse and the monitor
3. Use the HDMI cable for monitor connection
4. Use the USB Hub to connect both keyboard and mouse to your Raspberry Pi

Leeds Trinity University

Power up the Raspberry Pi

- Once you have completed connecting your Raspberry Pi to the mouse, keyboard and the screen, connect the USB power cable
- This will supply power to your Raspberry Pi, and it will turn on.
- Allow few minutes and you should shortly see Raspberry Pi OS starting to load on the screen.
- The next slide shows a sample of Raspberry Pie environment.


[pi@raspberrypi: ~]
14:15
Programming
Education
Office
Internet
Sound & Video
Graphics
Games
Accessories
Help
Preferences
Run...
Shutdown...
Add / Remove Software
Appearance Settings
Audio Device Settings
Keyboard and Mouse
Main Menu Editor
Raspberry Pi Configuration
Recommended Software
Configure Raspberry Pi system

Leeds Trinity University

Raspberry Pi

Raspberry Pi comes with its own operating system Pi OS previously called as *"Raspbian".*

By default on Model 3 of Raspberry Pi, you have connectivity options

- WiFi, Bluetooth

I/O Ports:

- HDMI, USB, GPIO Pins

Applications such as

- Thonny: Python IDE
- Web Browser
- Image Viewer
- Terminal

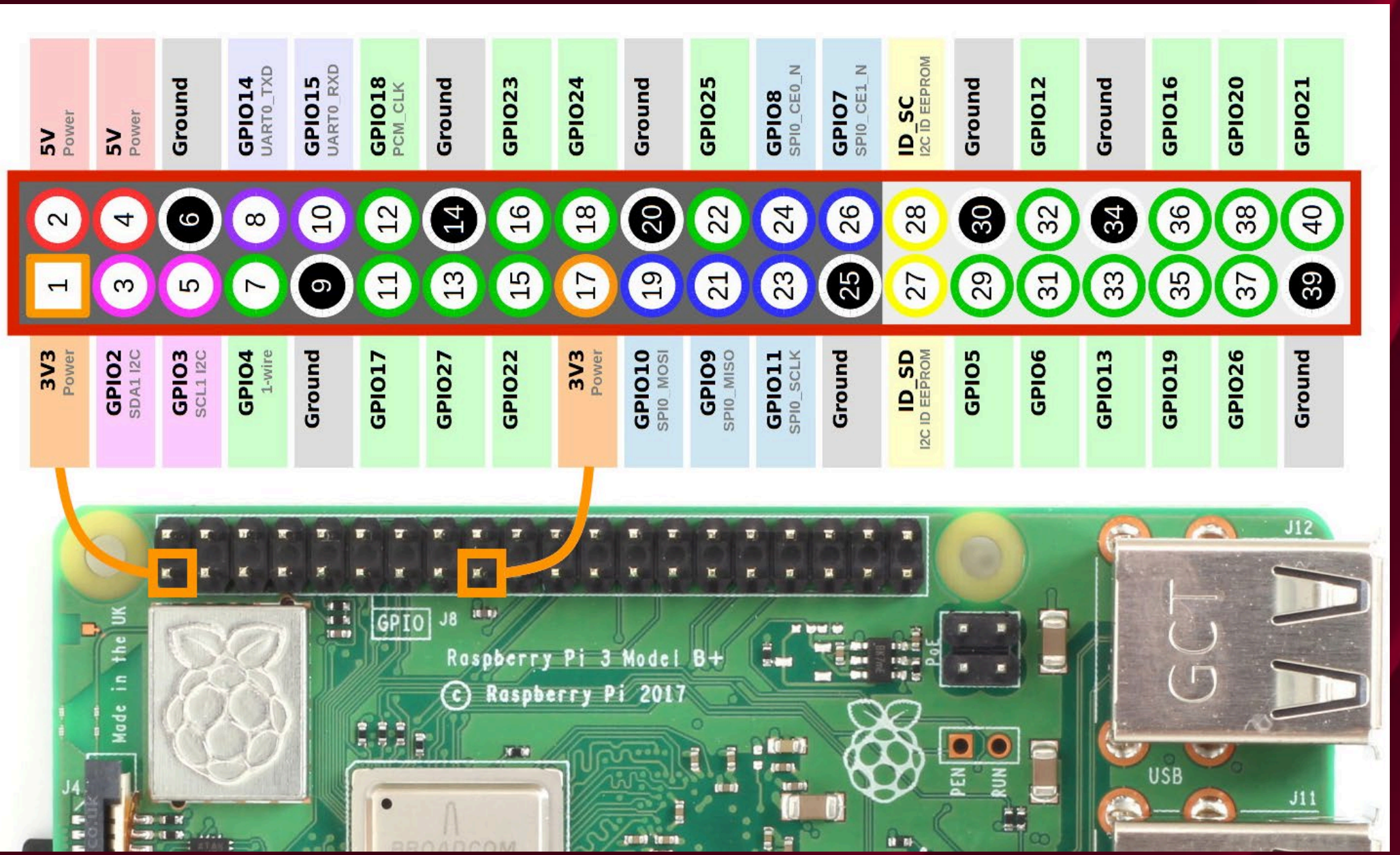
Leeds Trinity University

Task 2

Control an LED with Raspberry Pi

List of equipment that you will need or Task 1-3:

- LED
- Resistor
- Connections (Jumper wires: wires with sockets suitable for pin connections)
- A Raspberry Pi board
- Breadboard

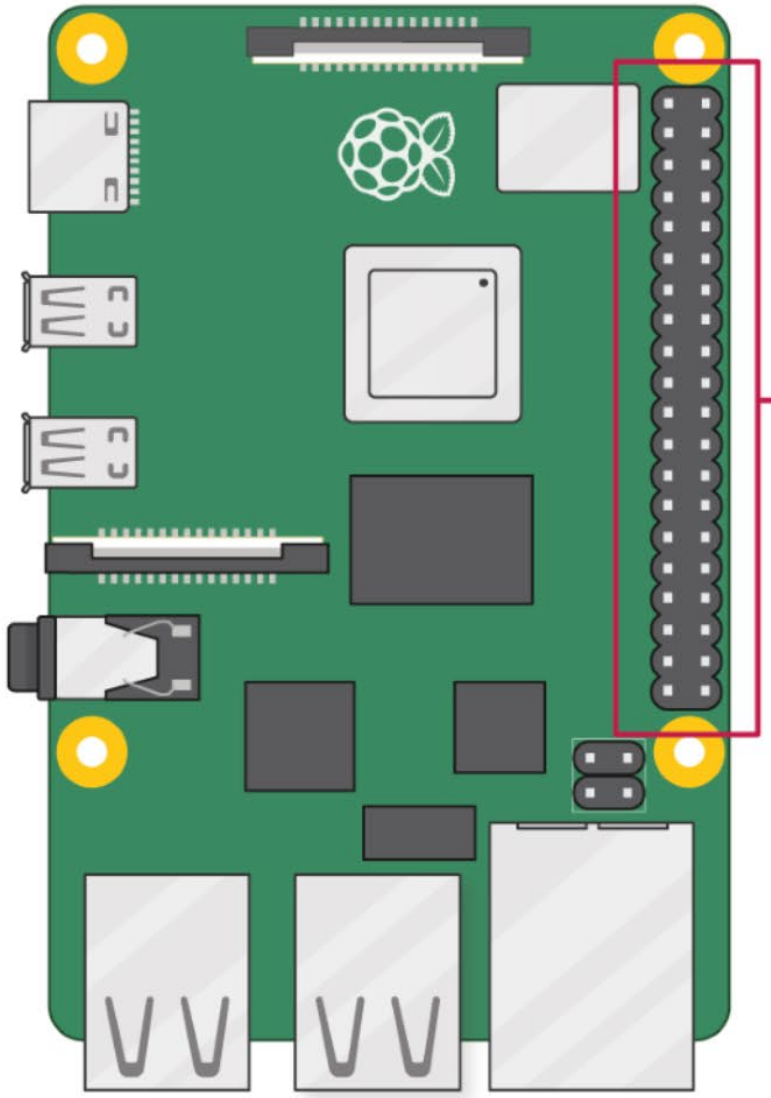


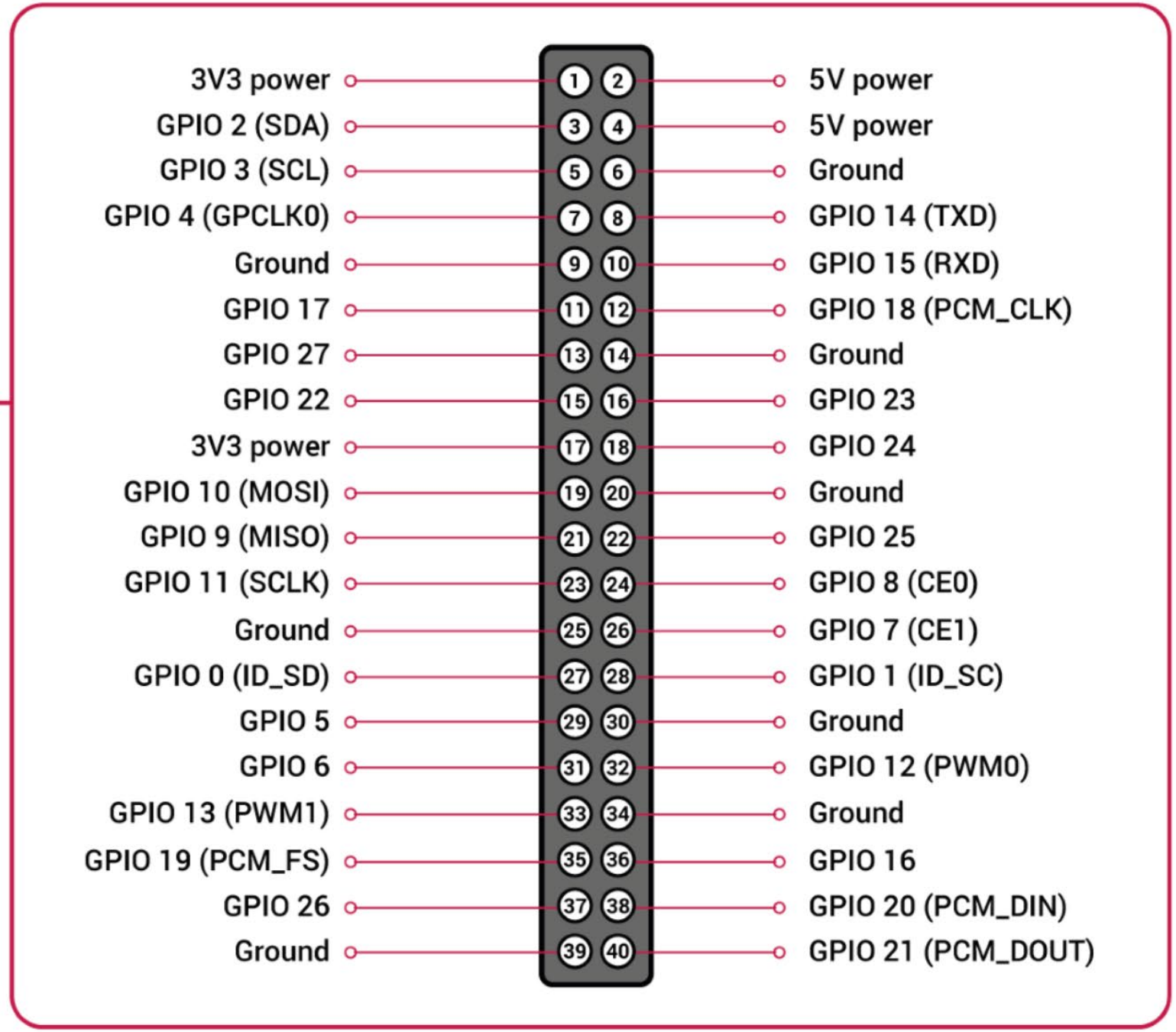
Pay attention to the locations of the Ground pins and Power pins in the image.

Identify the Ground pins and Power pins (5v and 3.3v) in the image and locate them on your Pi board.

If you confuse the GND (Ground pin) and Power pin, it can seriously or permanently damage your board or other components.

Ask for help if not confident.




3V3 power
GPIO 2 (SDA)
GPIO 3 (SCL)
GPIO 4 (GPCLK0)
Ground
GPIO 17
GPIO 27
GPIO 22
3V3 power
GPIO 10 (MOSI)
GPIO 9 (MISO)
GPIO 11 (SCLK)
Ground
GPIO 0 (ID_SD)
GPIO 5
GPIO 6
GPIO 13 (PWM1)
GPIO 19 (PCM_FS)
GPIO 26
Ground
1 2
3 4
5 6
7 8
9 10
11 12
13 14
15 16
17 18
19 20
21 22
23 24
25 26
27 28
29 30
31 32
33 34
35 36
37 38
39 40
5V power
5V power
Ground
GPIO 14 (TXD)
GPIO 15 (RXD)
GPIO 18 (PCM_CLK)
Ground
GPIO 23
GPIO 24
Ground
GPIO 25
GPIO 8 (CE0)
GPIO 7 (CE1)
GPIO 1 (ID_SC)
Ground
GPIO 12 (PWM0)
Ground
GPIO 16
GPIO 20 (PCM_DIN)
GPIO 21 (PCM_DOUT)

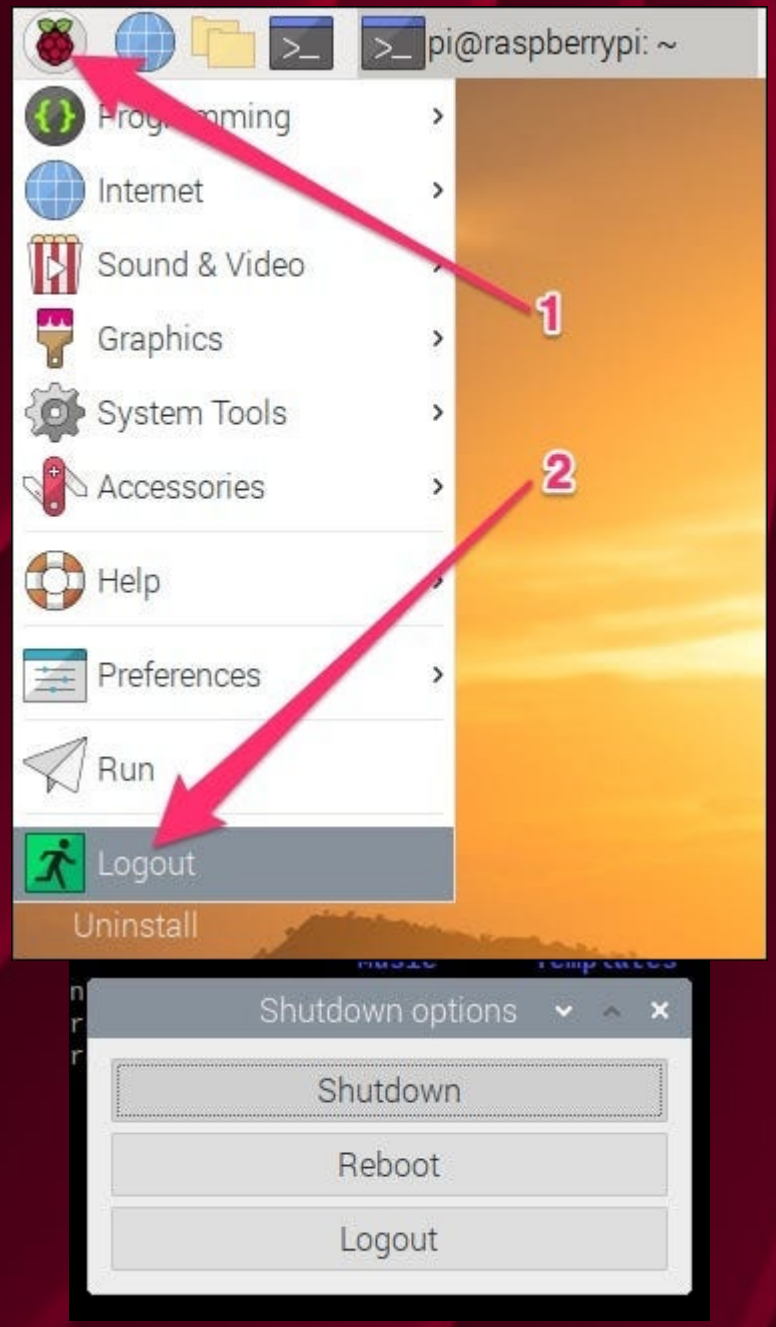
Leeds Trinity University

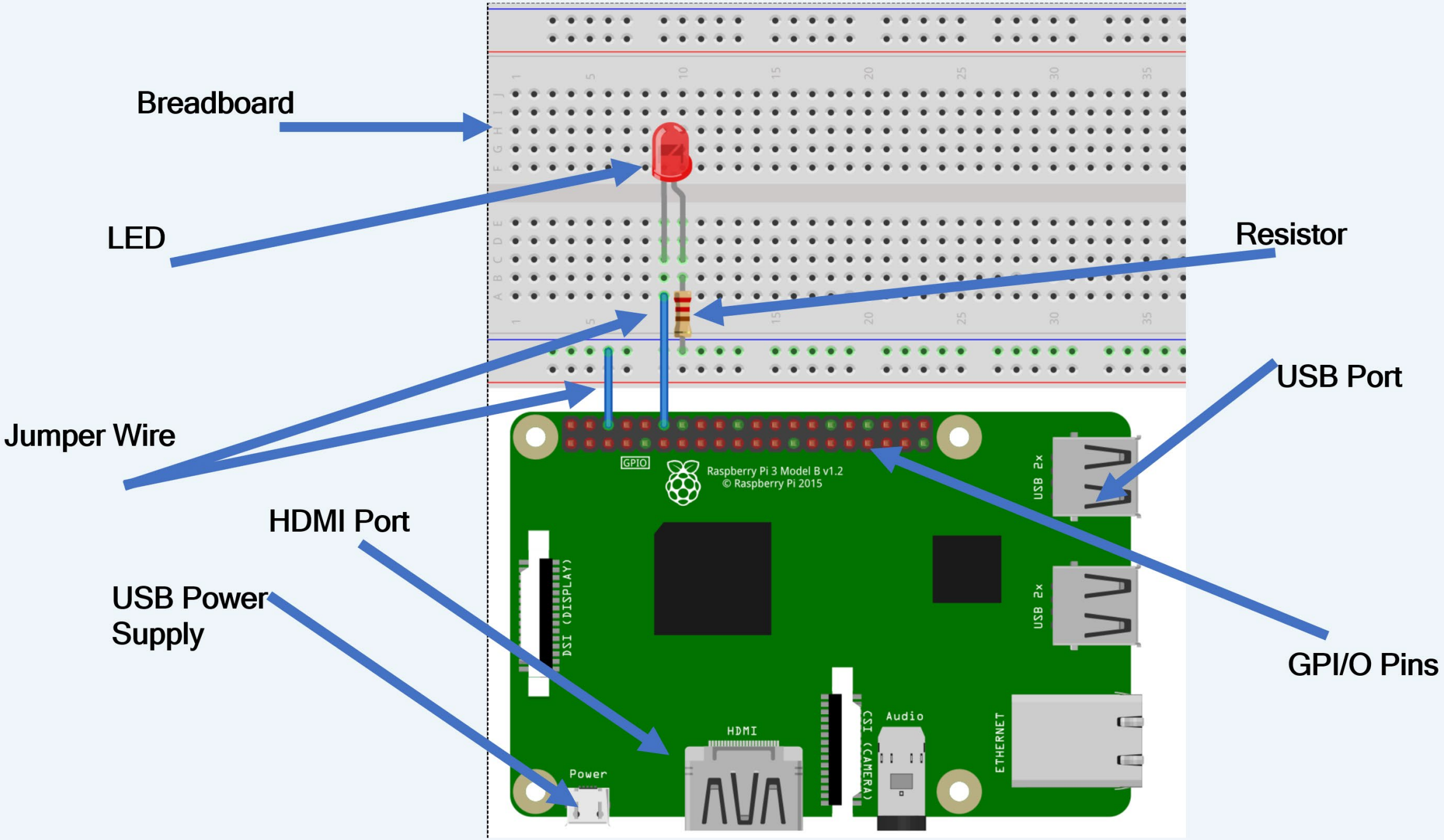
TURN OFF the POWER

1. Shut down the Raspberry Pi as demonstrated here.
2. Wait for a good minute.

3. Remove the USB Power Cable from Pi Board.

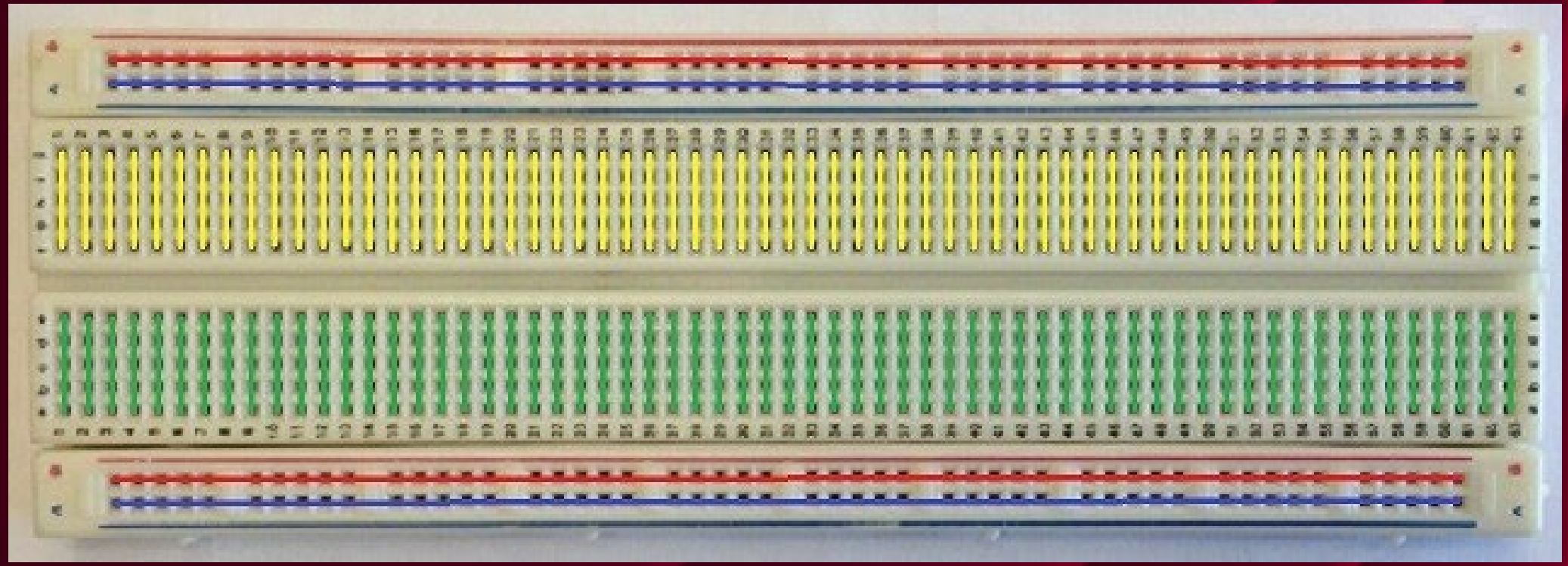
This is always needed before assembling the circuit and making connections.




Breadboard
LED
Resistor
Jumper Wire
USB Port
HDMI Port
USB Power Supply
GPI/O Pins
GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
Audio
CSI (CAMERA)
ETHERNET
Power

Leeds Trinity University

Breadboard Connectivity



- There are 4 blocks of pins demonstrated in the image of a breadboard above. All pins in each block are disconnected from other blocks.
- Pins highlighted with the same colour in each block is connected through out the board as demonstrated in the image.

Leeds Trinity University

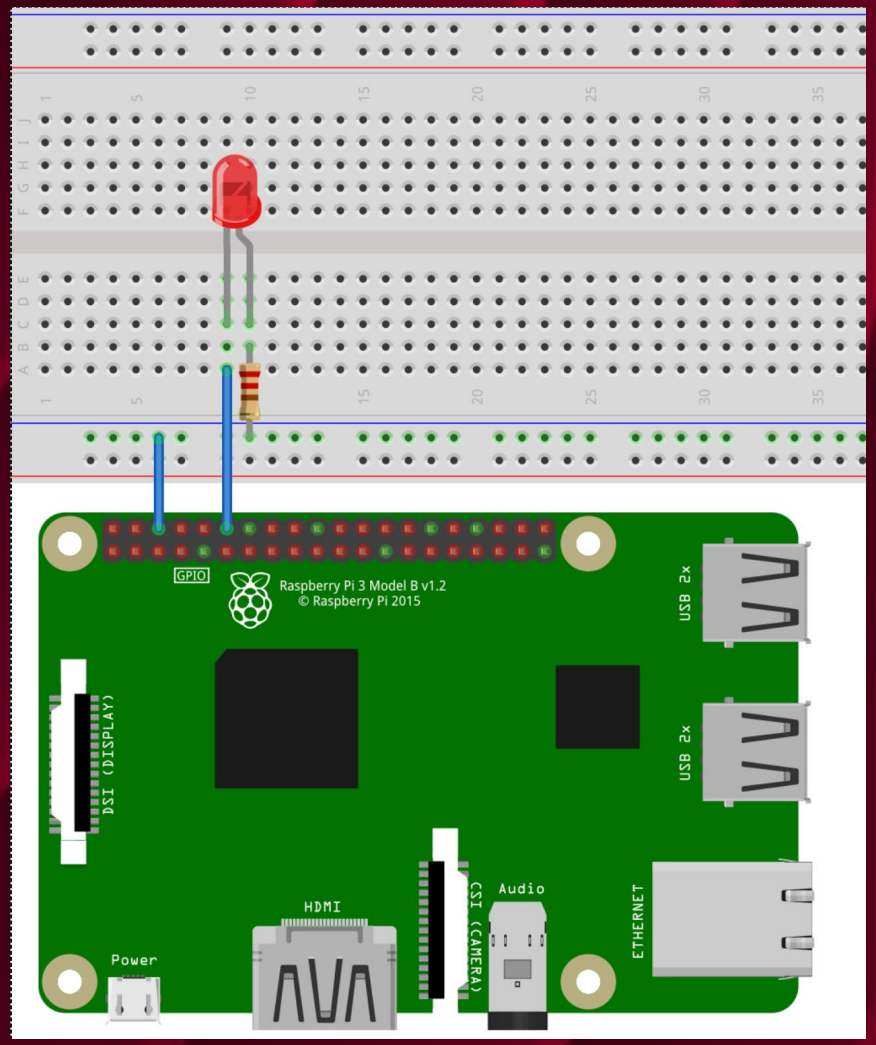
Circuit Assembly

- Make sure that there is no power supply connected to the Raspberry Pi

NO LED LIGHTS SHOULD BE ON/BLINKING

- Use one of the jumper wires to connect a Ground pin to the rail, marked with blue, on the breadboard. The female end goes into the Raspberry Pi's pin, and the male end goes into a hole on the breadboard.
- Then connect the resistor from the same row on the breadboard to a column on the breadboard, as shown here (also available in detailed steps in next slides)
- Next, push the LED Legs into the breadboard, you may need to switch the position of the long leg later on.

- Connect the pin18 to the right leg of the LED.

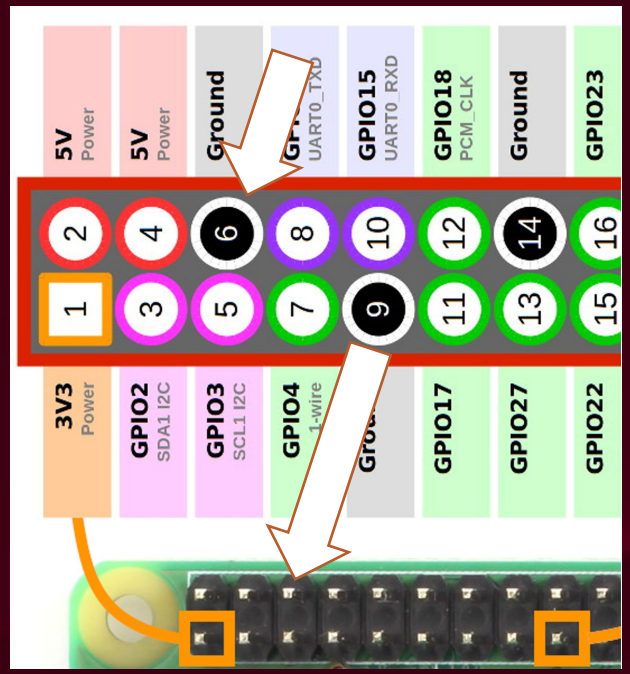


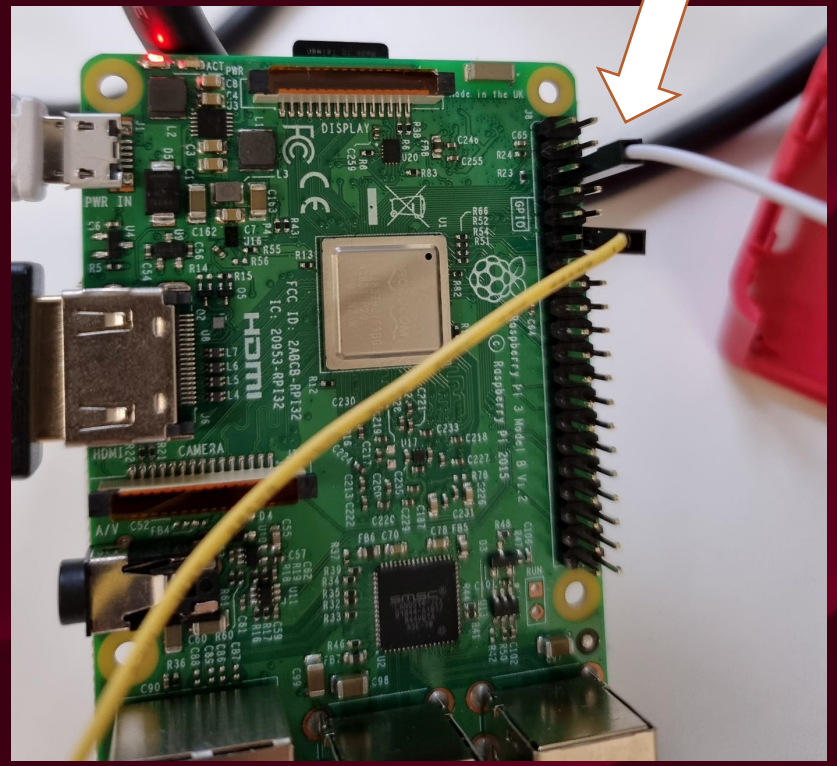
Circuit Assembly: Connect the GND

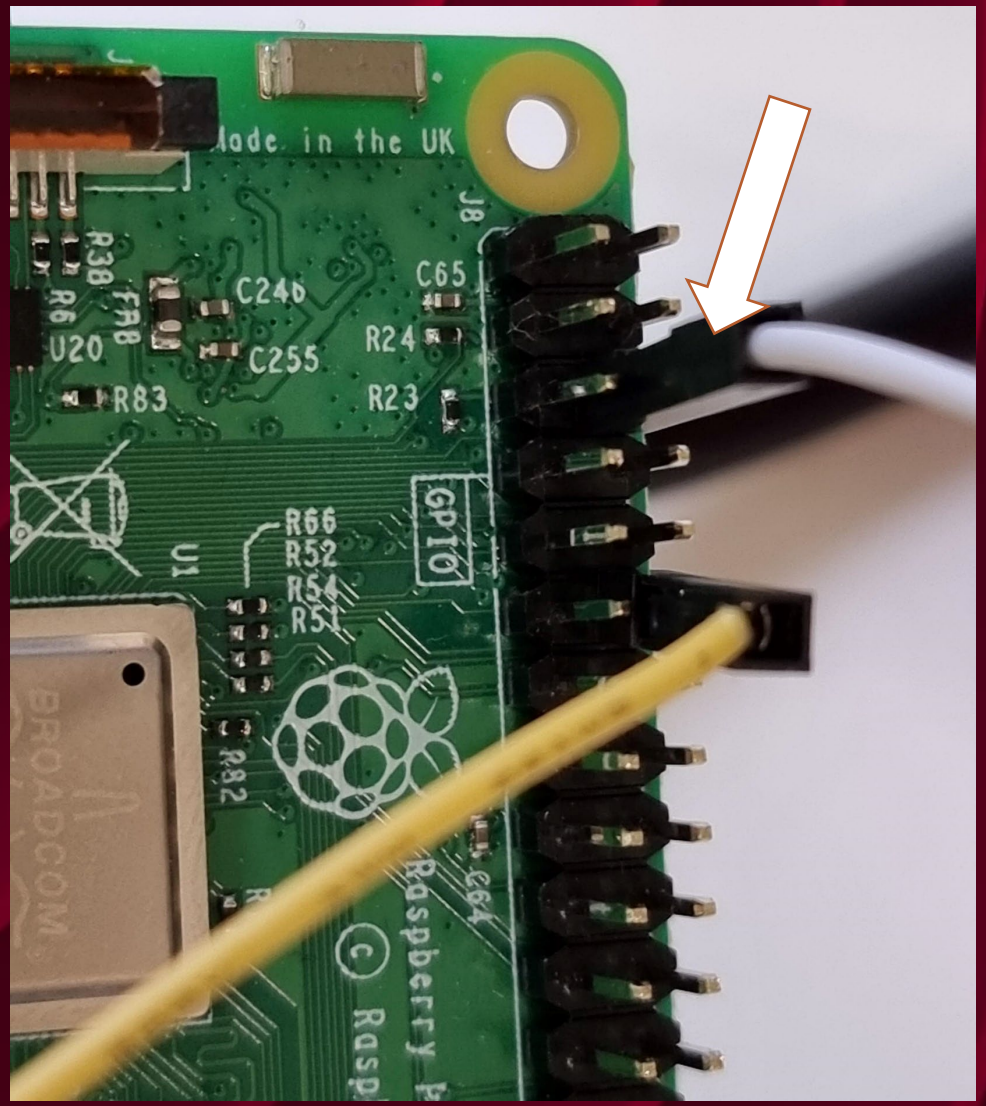
Leeds Trinity University

Go back to the Pin config of the Raspberry Pi, and find one of the GND pins.

We have used pin 6.





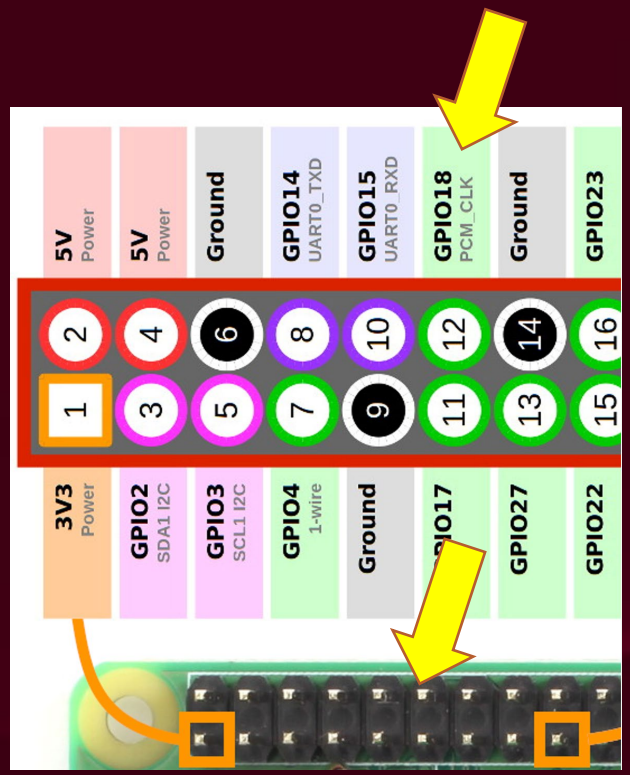


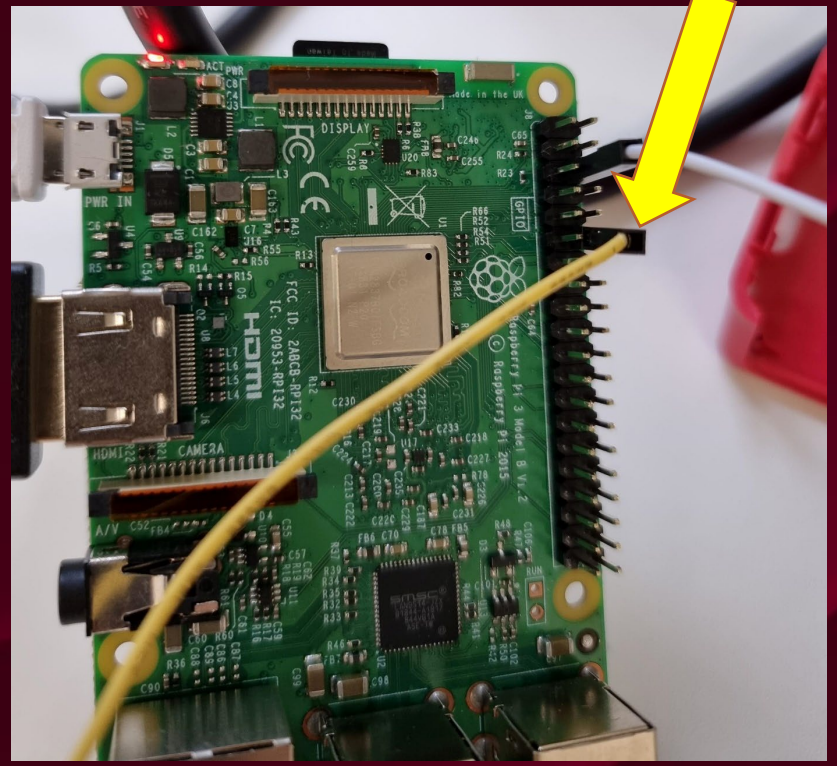
Leeds Trinity University

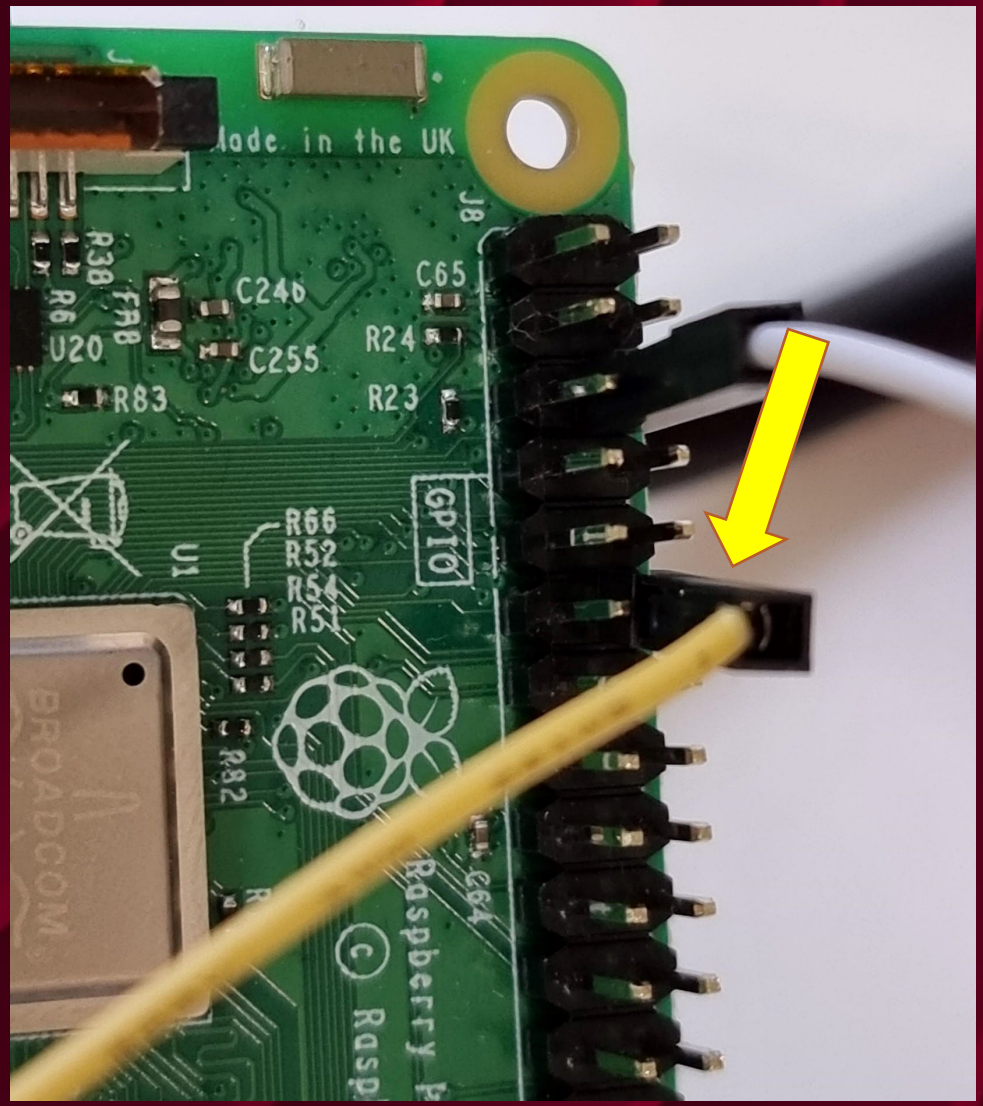
Circuit Assembly: Connect the GPIO 18

Go back to the Pin config of the Raspberry Pi and find GPIO 18.

Add a connector to the GPIO 18.



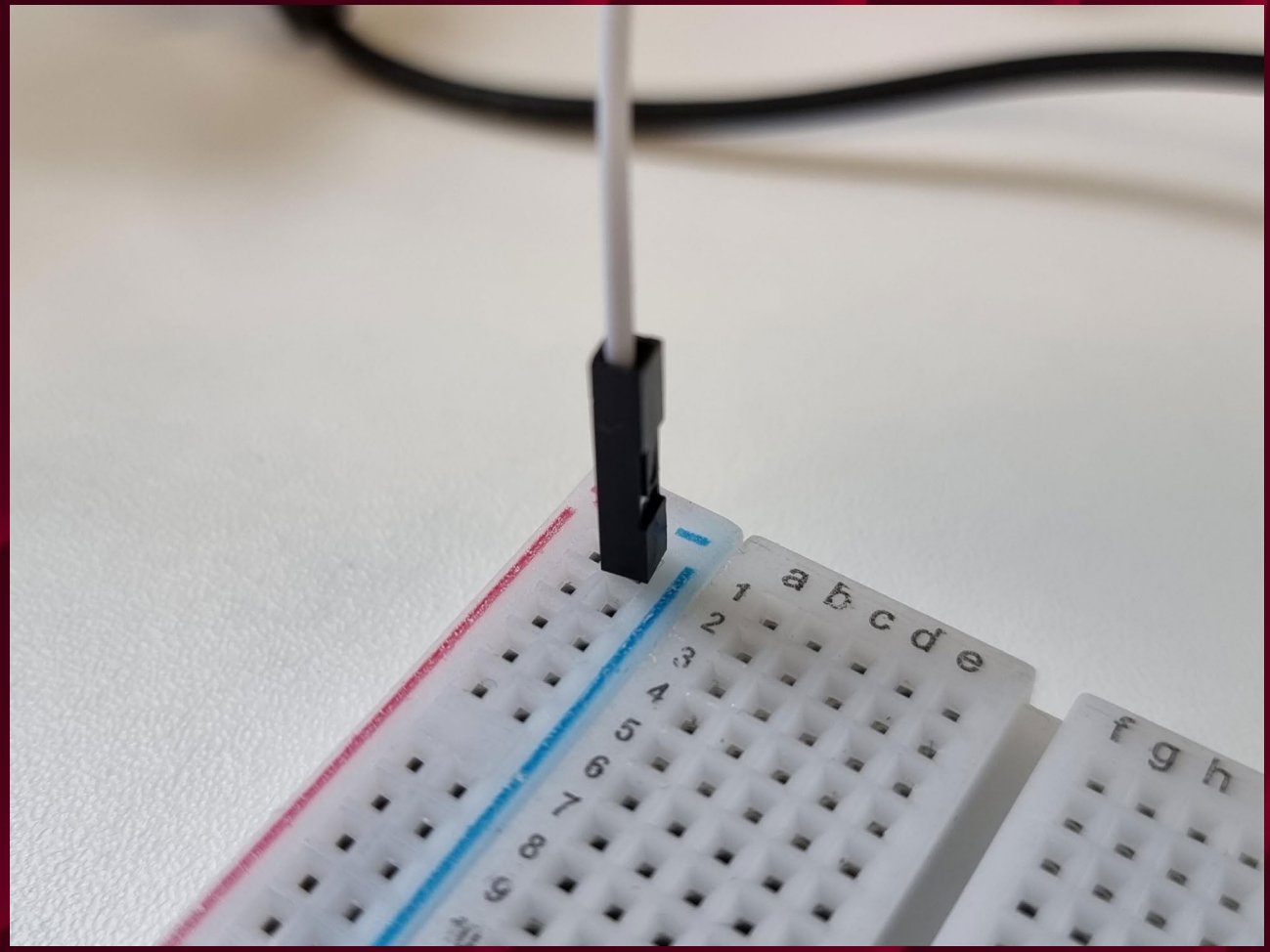




Leeds Trinity University

Breadboard Connections: Step 1

Connect the GND to the blue line (-ve) on the breadboard.



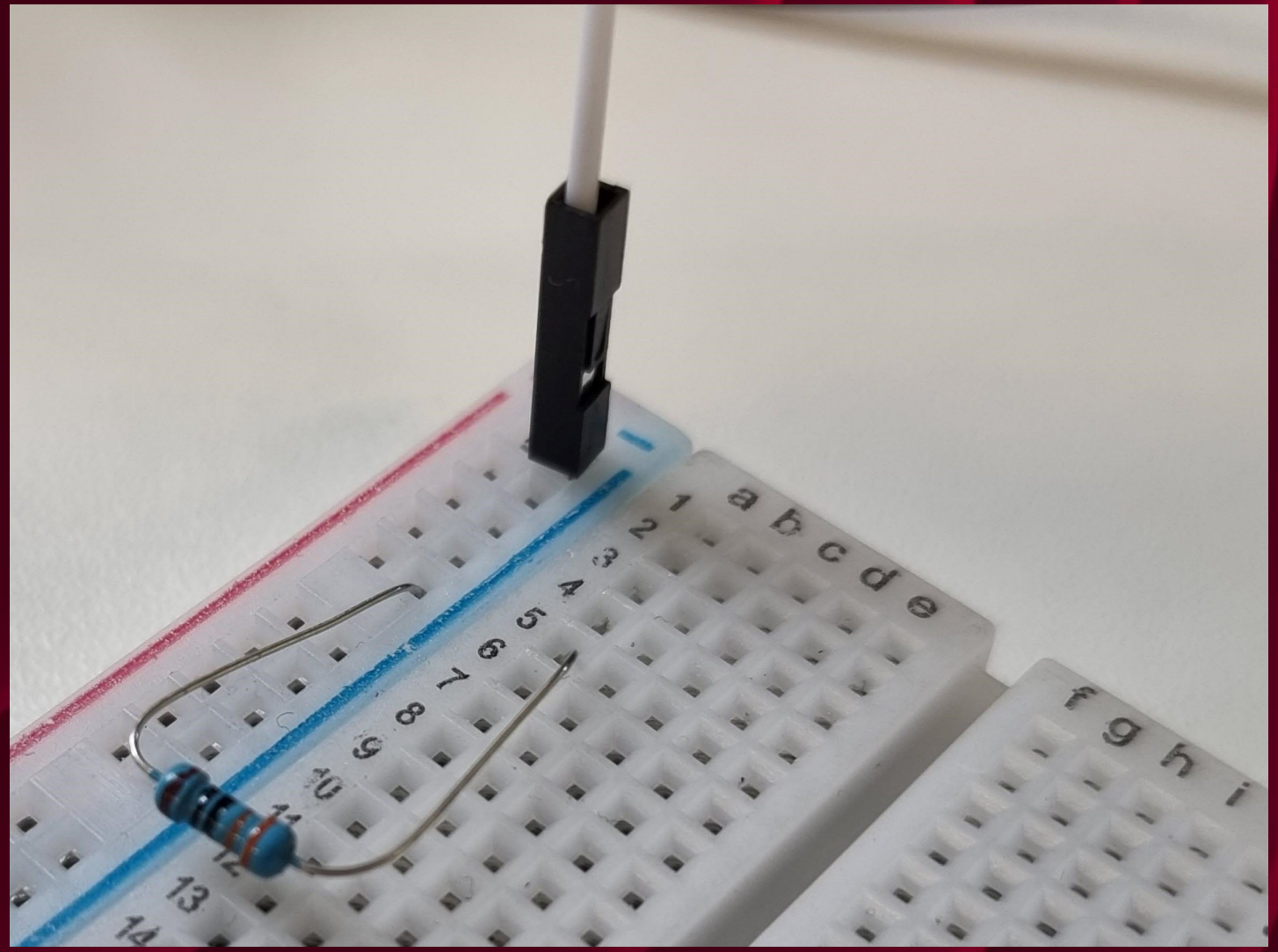
Leeds Trinity University

Breadboard Connections: Step 2

Add the resistor

It should go from any of the pins on the blue line to any of the pins on the block [a,b,c,d,e].

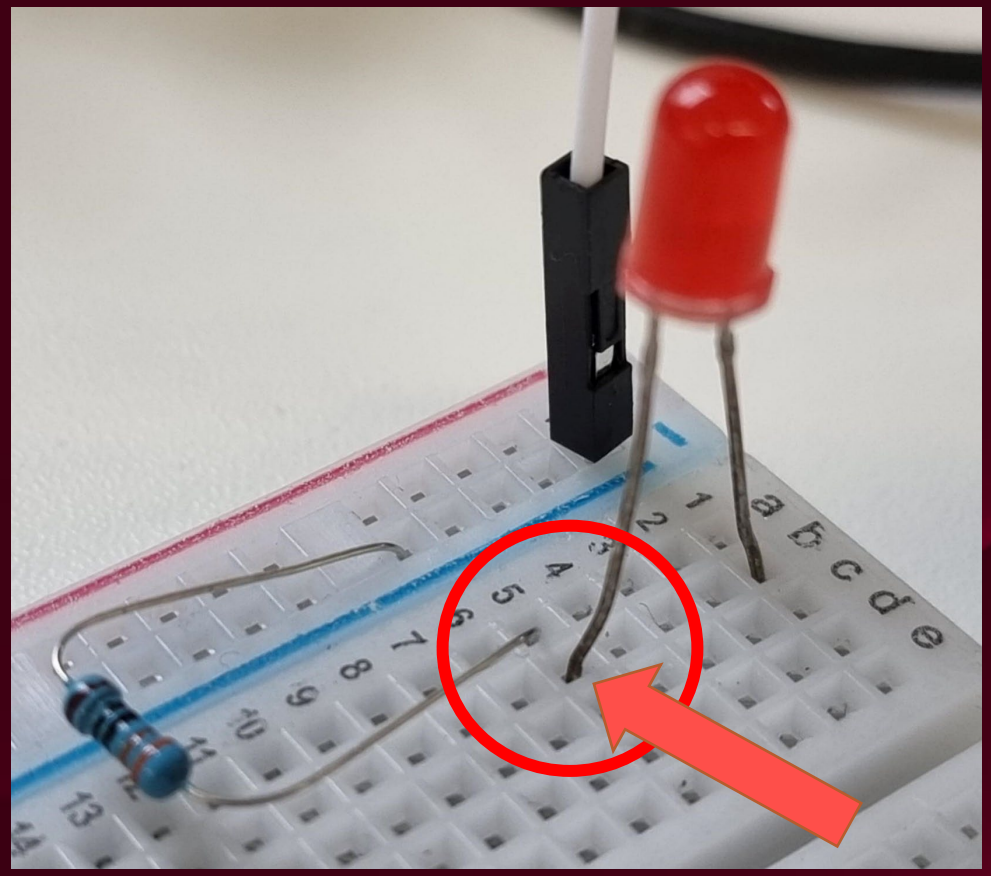
We have used a5.

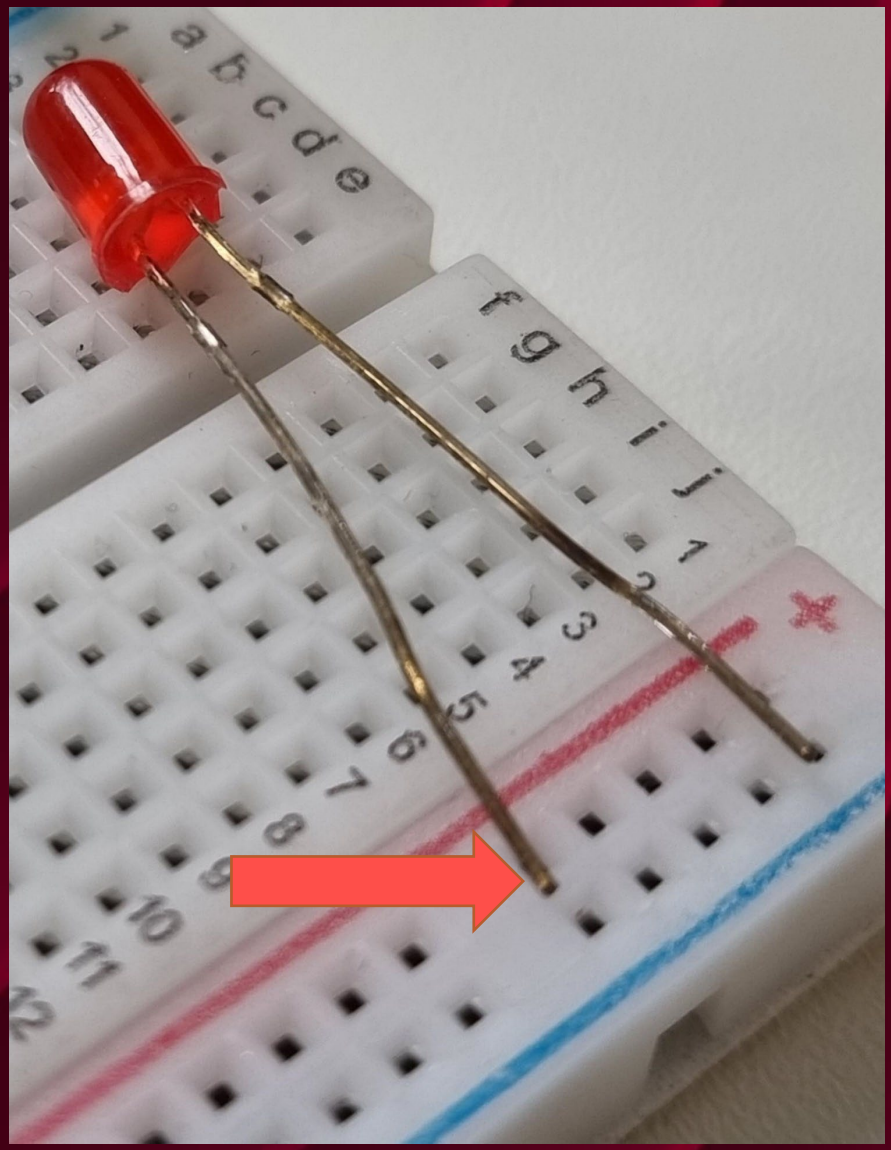


Leeds Trinity University

Breadboard Connections: Step 3

Make sure that the short pin of the LED is connected to the same column as the resistor.





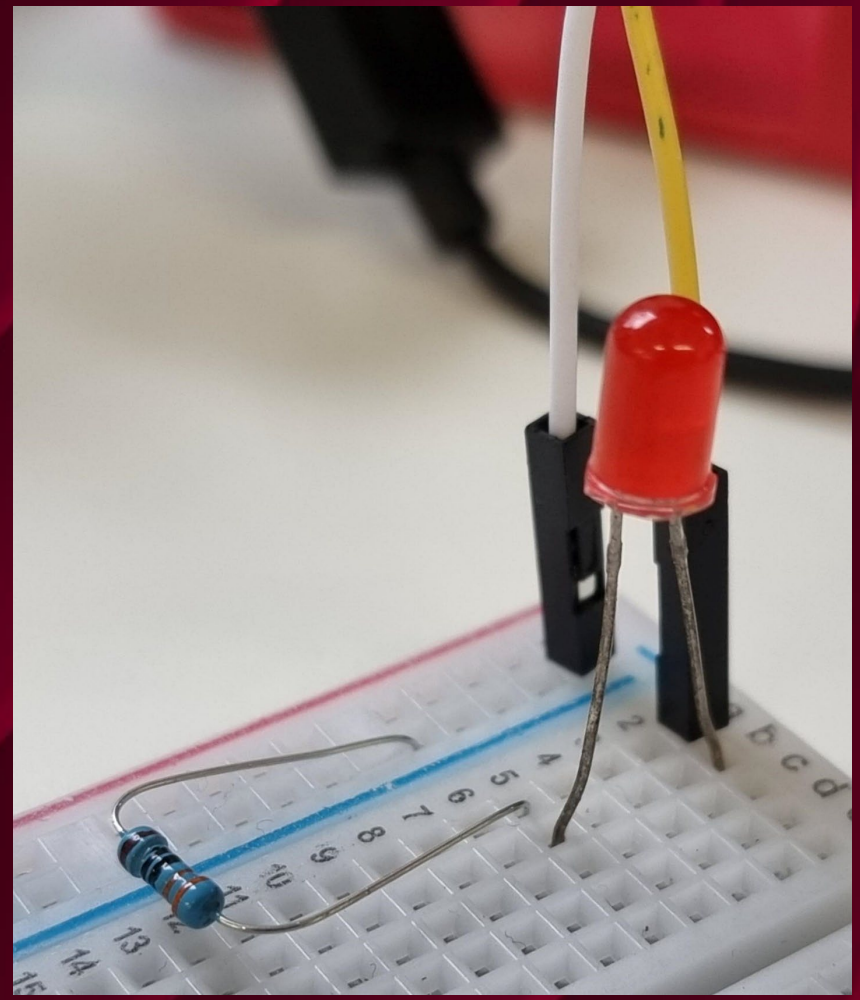
Leeds Trinity University

Breadboard Connections: Step 4

As the final step, connect the GPIO 18 to the same column as the LED.

We have gone with column 1.

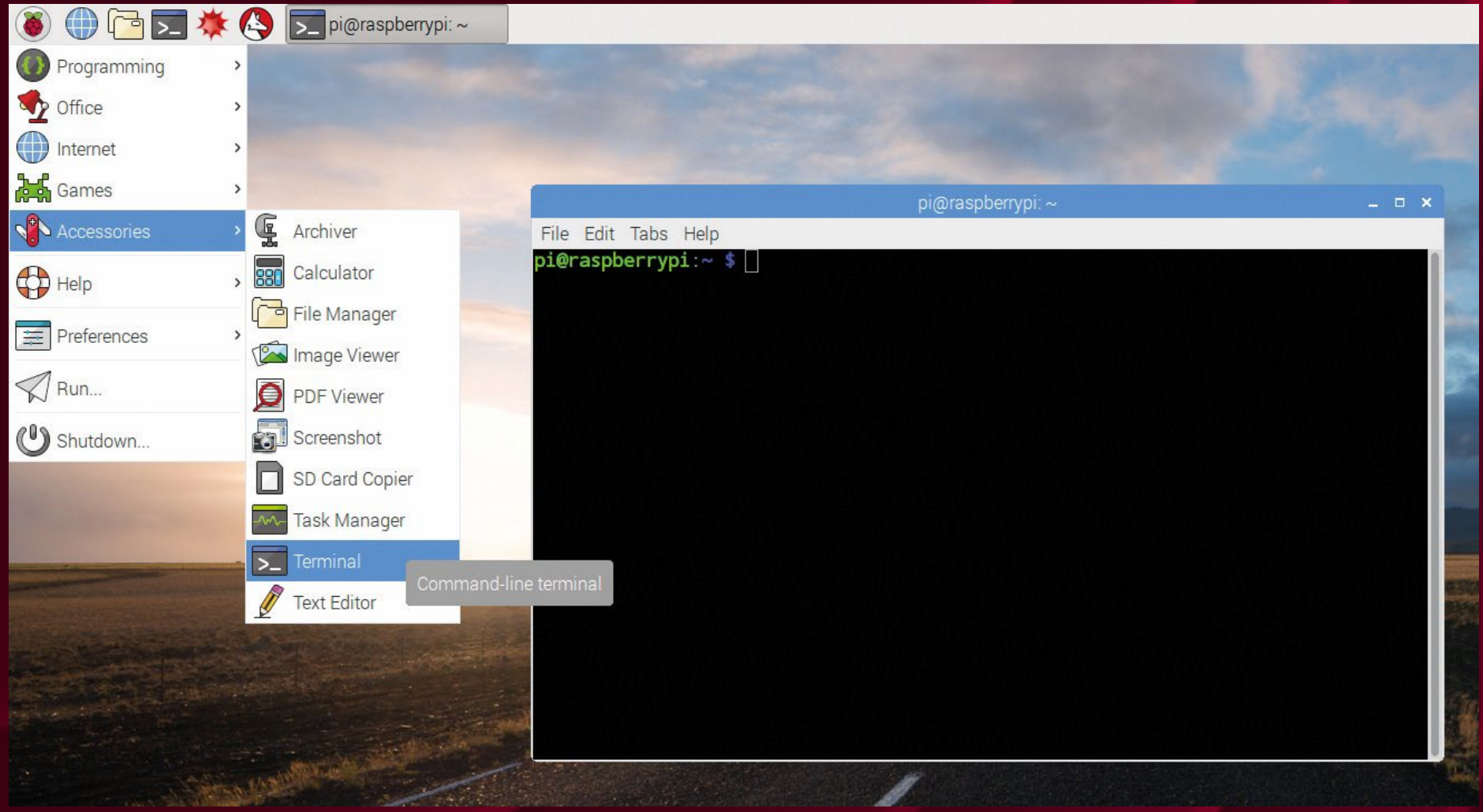
At this stage, wait for your lecturer to check your circuit.



Leeds Trinity University

Turn on the Raspberry Pi

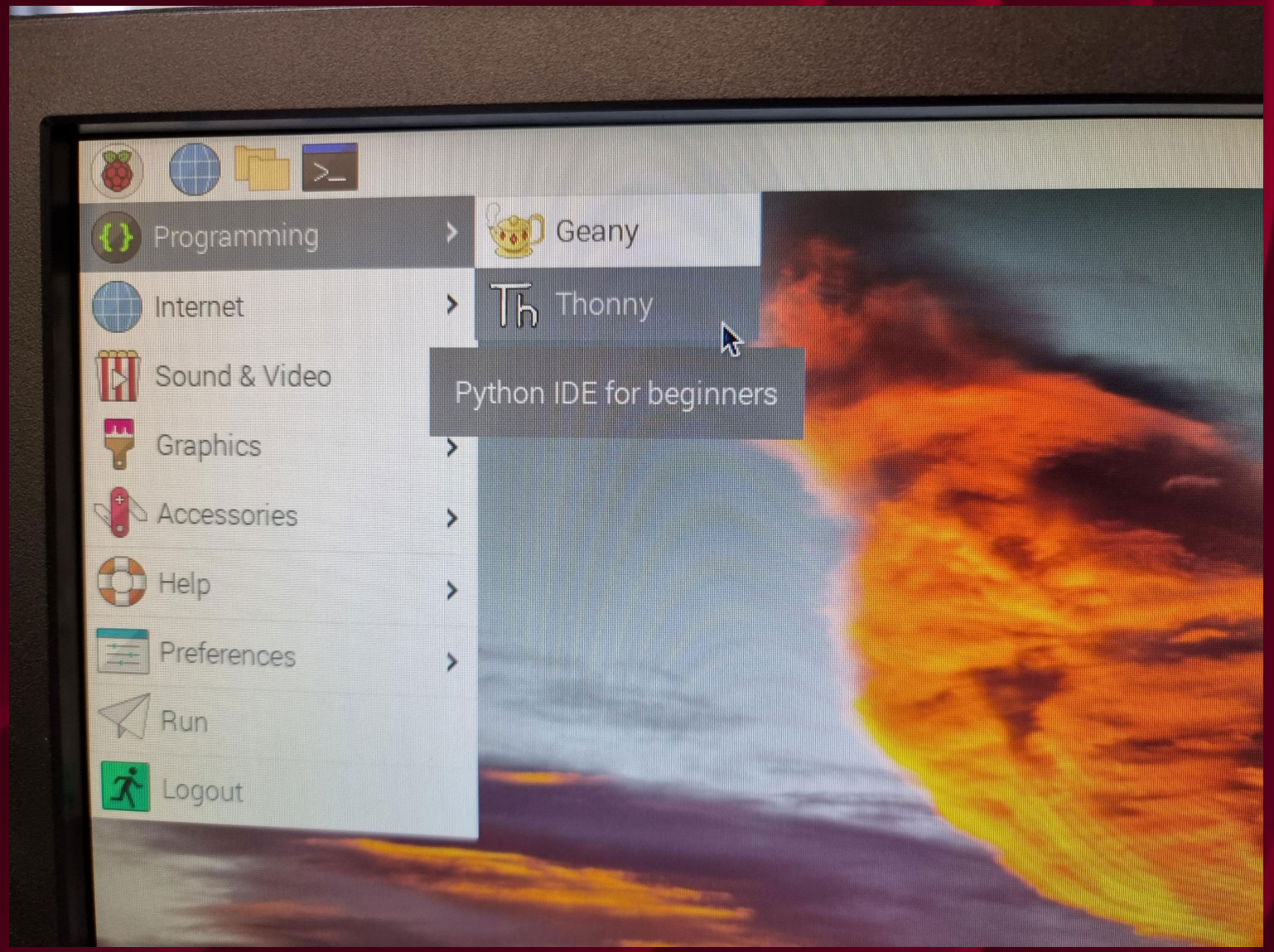
1. Connect the USB cable and turn on the Raspberry Pi
2. Open a Terminal Connection



Leeds Trinity University

Open and Run Thonny

1. Thonny is a Python IDE
2. It should be installed by default in your Raspberry Pi (if not you can install it from accessories)



Leeds Trinity University

GPIO Pins

- General Purpose Input Output (GPIO) pins provide a way for Raspberry Pi to interface with the outside world by connection to external electric circuits.
- The picture below shows the arrangement of the GPIO pins on the model of Raspberry Pi model 3B.
- Pins marked GND are ground pins (zero-volts).

Leeds Trinity University

Sample Code 1: Turn On/off an LED

- Create a text file and replicate the following snippet
- Save it as LED.py (.py is the extension for Python files)

Or check the code within the documents folder.

Make an attempt to read the code and guess how it works.

Write down your answer in a word document and submit a screenshot to Padlet.

https://bit.ly/ohdpdlet

```
1  import RPi.GPIO as GPIO
2  import time
3  GPIO.setmode(GPIO.BCM)
4  GPIO.setwarnings(False)
5  GPIO.setup(18,GPIO.OUT)
6  print("Light on")
7  GPIO.output(18,GPIO.HIGH)
8  time.sleep(1)
9  print("Light off")
10 GPIO.output(18,GPIO.LOW)
```

Leeds Trinity University

Sample Code 2: Blink an LED

- Create a text file and replicate the following snippet
- Save it as LED_Blink.py (.py is the extension for Python files)

Or check the code within the documents folder.

Make an attempt to read the code and guess how it works.

Write down your answer in a word document and submit a screenshot to Padlet.
https://bit.ly/ohdpdlet

```
1   import RPi.GPIO as GPIO
2   import time
3   GPIO.setmode(GPIO.BCM)
4   GPIO.setwarnings(False)
5   GPIO.setup(18,GPIO.OUT)
6   for i in range(6):
7       print("Light on")
8       GPIO.output(18,GPIO.HIGH)
9       time.sleep(1)
10      print("Light off")
11      GPIO.output(18,GPIO.LOW)
12      time.sleep(1)
```

Task 3
Control the LED Light from a WebApp

We will develop a web app using Flask and Python in this task.

Leeds Trinity University

- Create a text file and replicate the following snippet
- Save it as route.py (.py is the extension for Python files)

```python
# import some functions and classes from external packages
from flask import Flask, render_template
import RPi.GPIO as GPIO
import time

# tell program what naming convention to use to identify I/O pins
GPIO.setmode(GPIO.BCM)
# suppress GPIO warninga
GPIO.setwarnings(False)
# tell Python that pin 18 will be used to output information
GPIO.setup(18,GPIO.OUT)

# initialuse the web app
app = Flask(__name__)

```

```python
# define some 'routes' or 'views' in the app
@app.route('/')
def home():
    return render_template('index.html', status='off')

@app.route('/on')
def on():
    # Task 1b: put code here to turn light ON
    GPIO.output(18,GPIO.HIGH)
    return render_template('index.html', status='on')

@app.route('/off')
def off():
    # Task 1b: put code here to turn light OFF
    GPIO.output(18,GPIO.LOW)
    return render_template('index.html', status='off')

if __name__ == "__main__":
    app.run()
```

Leeds Trinity University

HTML & CSS

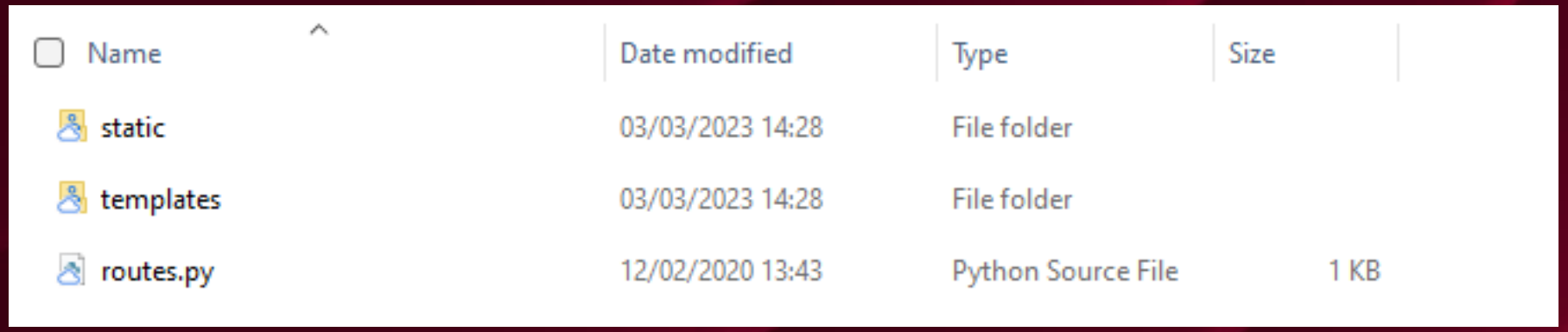
HTML: Hypertext Markup Language

Create a text editor and replicate the HTML and CSS code in the next slides.

The code is also available from the GitHub Classroom/Google Drive.

The HTML file (index.html) should be placed within the “templates” folder next to the python file developed earlier.

The CSS file should be placed in a folder named “static”.



Leeds Trinity University

HTML: Web Page

- Create a text file and replicate the following snippet
- Save it as index.html (.html is the extension for HTML files)

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
        <title>MyHome Dashboard</title>
    </head>
    <body>
        <div id="outer">
            <h1>MyHome Dashboard</h1>
            <!-- Task 2: you can edit the Link text if you like! -->
            {% if status=='on' %}
            <a href="{{ url_for('off')}}" id="lightOff" class="button">Light Off</a>
            {% else %}
            <a href="{{ url_for('on')}}" id="lightOn" class="button">Light On</a>
            {% endif %}
        </div>
    </body>
</html>
```

Leeds Trinity University

CSS Snippet

- Create a text file and replicate the following snippet
- Save it as style.css
- The css file should be saved within the folder static/css/style.css
- Follow the line numbers
- Code available from Google Drive and GitHub Classroom as well

```
1  #outer {
2      text-align: center;
3      padding: 20px;
4  }
5
6  h1 {
7      font-family: helvetica, sans-serif;
8      font-size: 24px;
9      margin: 1em 0;
10 }
```

```
11
12 a.button {
13     display: block;
14     width: 100px;
15     height: 60px;
16     margin: 0 auto;
17     padding-top: 40px;
18     border: 1px solid rgb(10, 10, 200);
19 }
20
21 a#lightOff {
22     color: black
23     background-color: IndianRed;
24
25 }
26
27 a#lightOn {
28     color: black;
29     background-color: g een;
30 }
```

Task 4: Plan a pitch

For this final task, plan a 2-minute pitch to answer the following questions:

Pitch to

- What will your app do?
- Who may use the app? Who is it useful for?
- How will they benefit from using it?
- Can you think of applications of the app in real world?

Leeds Trinity University

A Gift from LTU

It was nice meeting you in the Offer Holders Day today.

Here is a small gift to take away with you and you can continue to experiment and enjoy working with a Raspberry Pi.

There are loads of projects available from Raspberry Pi Foundation
Projects | Computer coding for kids and teens | Raspberry Pi

Leeds Trinity University

Make More Happen

Thank you.
Any questions?

Contact details:
Dr Yashar Baradaranshokouhi
Yash.b@leedstrinity.ac.uk

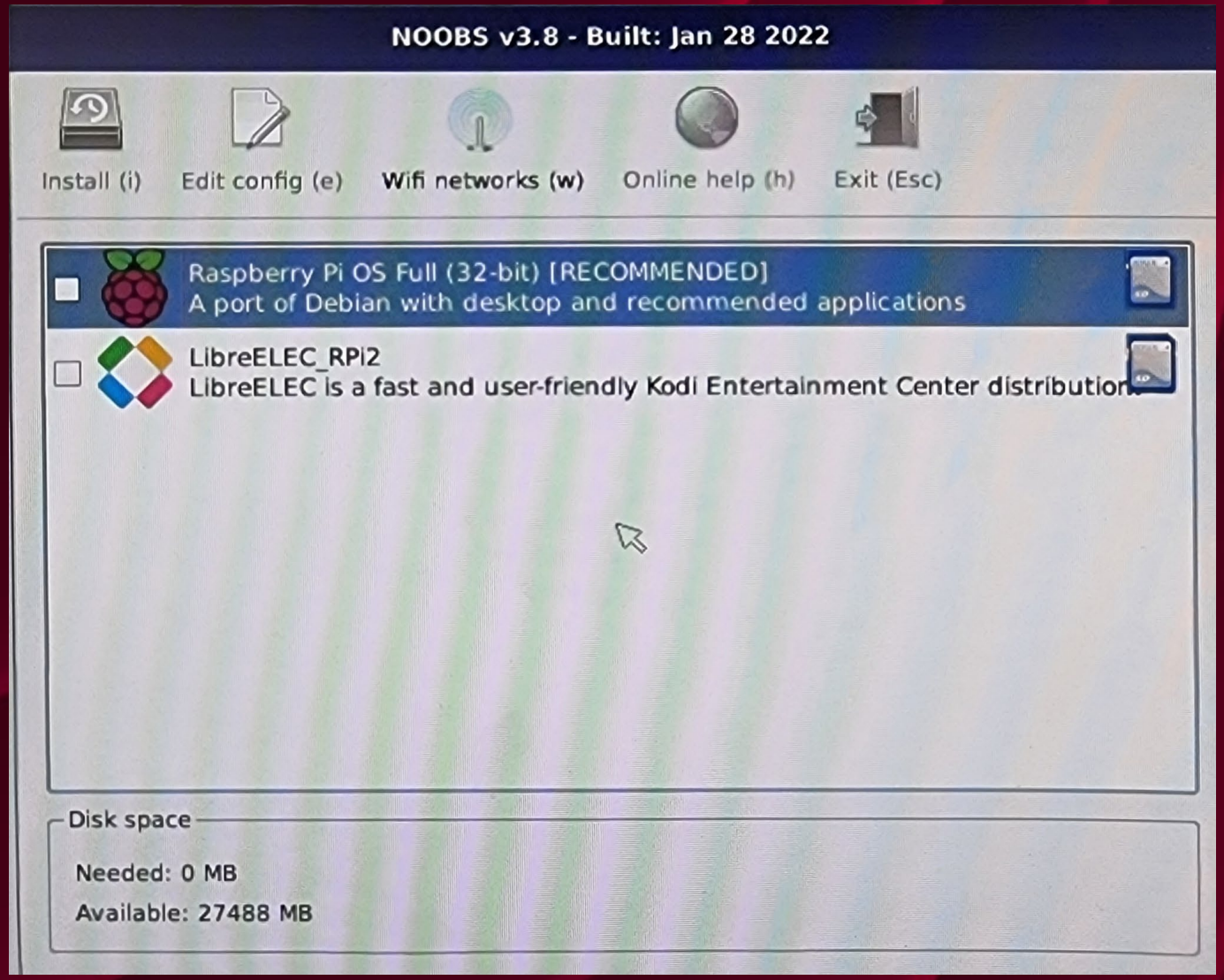
Leeds Trinity University

Pi OS Installation

When you connect your R Pi for the first time

- You will need to select Raspberry Pi OS Full (32-bit) [Recommended]
- Then click on install.

You can also connect to WiFi and additional options will be available for installation (including a 64-bit Pi OS).



Leeds Trinity University

Pi OS Installation

Installation should take up to 30 minutes, and then you can log-in to your Raspberry Pi OS and start working with it.