Useful Information For Running At Home

Equipment required to complete a tile:

- NodeMCU processor (www.bitsbox.co.uk)
- Battery pack (3 AA's)
- Prewired LED's + cable connector strips + connecting wires (f/f jumpers)

Download tiles software onto NodeMCU: www.warwick.ac.uk/tilesfortales/hardwareandsoftware (Already completed on your chip)

Configuring Tile on home network

The NodeMCU acts as a wireless access point if it can't find the tiles network when the NodeMCU is powered up. Usually its ssid starts with ESP followed by numbers.

1. Connect to the NodeMCU network Using your computer connect to the nodeMCU wireless access point. The password is: tiles-for-tales
2. Load Config file Startup your browser and navigate to: http://192.168.10.1/config.html This loads up a page to enter your node name (for ScratchX), the home ssid and password.
3. Enter node name and home Wi-Fi parameters On the config page enter your node name, home ssid and password. This will mean the NodeMCU will join your home network and will broadcast its own IP address every 30 seconds.
4. Find NodeMCU IP address There are two ways of finding the NodeMCU IP address, one is to use a UDP app on a mobile phone to monitor broadcasts. The second way is to look at the wifi router and look for an ESP device.
5. Control tile using ScratchX Once you've established the NodeMCU's IP address on your home network, you can control it via the ScratchX website, first load esten-sion_loader, and use it to load your tile commands.

A bit more about the Technology Volunteers...

The Technology Volunteers are a student led group of volunteers at the University of Warwick. Each year student leaders recruit new volunteers from a range of departments across the university. Volunteers go into local schools in teams to run sessions on Arduino and Scratch. The aim is to encourage creators rather than consumers and the project is now in its 8th year.

Tiles For Tales

Who are we?

The Tiles for Tales project is a collaboration between Margaret Low of University of Warwick, James Johnston, and the student leaders of the Warwick Technology Volunteers. @megjlow @techvolunteers

What are we doing?

Exploring links between the craft of story telling and technology. This project shows how physical computing can be integrated into creative collaborative activities.

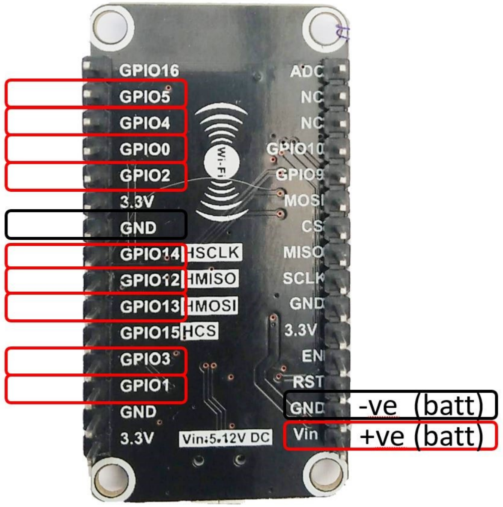
This project is based around the concept of story stones or story dice which can be used to tell or create a story. Participants work in groups to create and program a tile. These tiles can then be brought back together to make a 'quilt' of tiles which can be activated in a predetermined or a 'random' sequence.

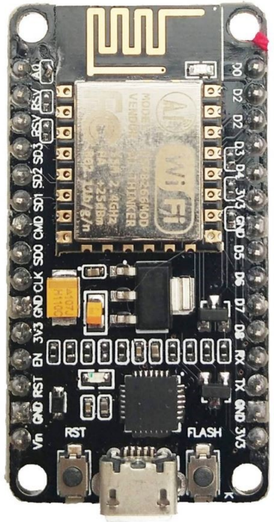
What is a tile?

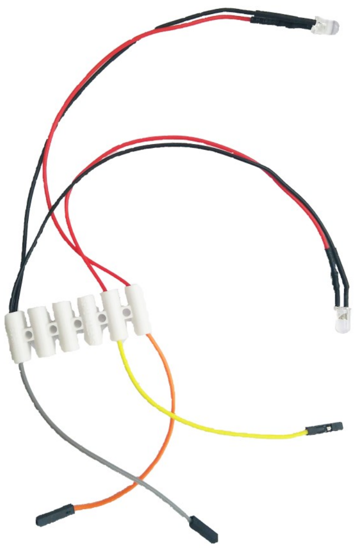
- A tile represents a scene or character
- Can use different materials, e.g. felt, paper
- Each tile is controlled by a Scratch extension
- It's an individual component and can be part of a larger installation

Contents of Box

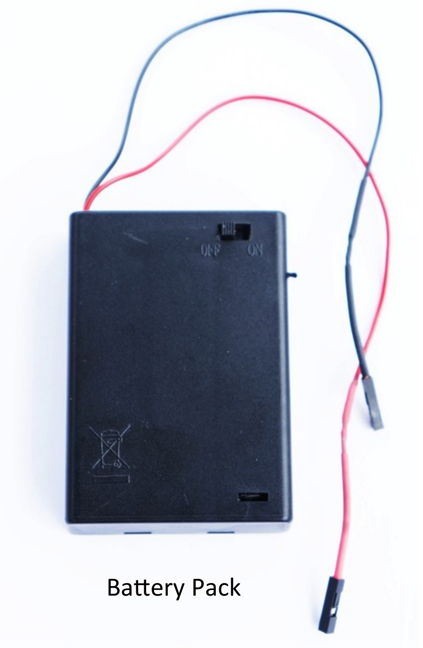
NodeMCU







LED Wiring harness



Battery Pack

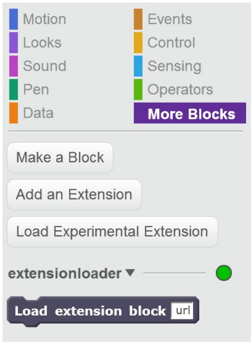
Software

We're using a ScratchX extension in order to import blocks to control a tile. How to load the extension:

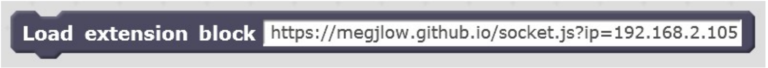
1. Using chrome browser, go to the ScratchX extension page: www.scratchx.org

2. There is a problem passing arguments into ScratchX. We've created a simple extension 'loader' which 'gets round' this issue. This extension loader can be loaded by clicking on '**Open extension URL**' and pasting the URL: https://megjlow.github.io/extensionloader.js

3. This creates a ScratchX block to load further extensions, with arguments.



4. Pull the '*Load extension block*' into the scripts https://megjlow.github.io/socket.js?ip=192.168.2.105 (replacing the IP address with the IP address of your NodeMCU). This loads the extension with commands to control the pins on your board:

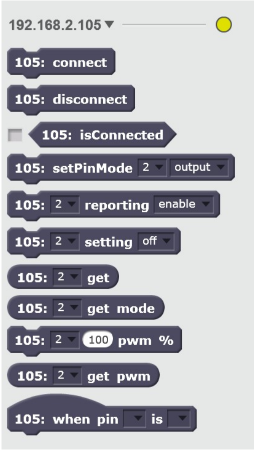


Load extension block https://megjlow.github.io/socket.js?ip=192.168.2.105

5. Click on the 'Load extension' block to run it, this will load in the extension with commands that enable you to control the NodeMCU pins:

6. Multiple NodeMCU extensions can be loaded via this method.

7. Many of the NodeMCU digital pins can be controlled via this extension. They can be set as input, output or pwm.



192.168.2.105
105: connect
105: disconnect
105: isConnected
105: setPinMode 2 output
105: 2 reporting enable
105: 2 setting off
105: 2 get
105: 2 get mode
105: 2 100 pwm %
105: 2 get pwm
105: when pin is