Twisted Sunrise Alarm.

Using Node-RED, Raspberry Pi Zero 2 W, Amazon Alexa, SG90 Servo.

The purpose of this device is to wake up the end-user up at sunrise and punish them if they do not, through pushing a mobile device on to a hard surface below.

This is made possible by using Node-RED with the Sunrise-Sunset.org API and Voice monkey API with Amazon Alexa Routines, with the Servo (SG90) controlled by a

Repo: https://github.com/Jaxku/TwistedSunriseAlarm/tree/main

Example video: https://youtu.be/cbCQWEUXY9Y

Bill of Materials:

Item	Purpose	Notes/Cost
Raspberry Pi Zero 2W	Main Controller	Long wait if the item is out of stock. Cost: $36.80 (PBTech)
MicroSD Card	OS/Node-RED For Pi	Minimum 32GB at a write speed of 30MB/s (V30) Cost: 15.36 (PBTech)
Power Supply	Powering the Pi	Use any 5w adapter (such as an Apple adapter) Or $16-17 for an official adapter at PBtech.
Tapo Smart Plug (P100)	Controls Lamp	Already owned
Wi-Fi Network (Support for 2.4GHz band)	Connectivity	Minimum speed of 10Mbps (Enough for IoT packets and Lossless audio streaming)
Echo Dot / Echo Show	Speaker, Manual Override	$100-250 (Depending on model)

Instructions if you are using the device yourself:

Setup your API:

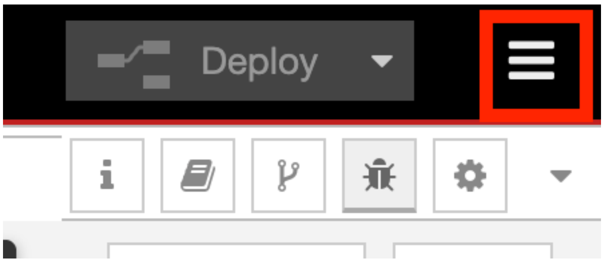
For Node-RED to know when to trigger the Servo-motion you will need to collect this information from a weather source in a format that is understandable to Node-RED.

You can install Node-RED on a Raspberry Pi running Raspberry PI OS and install Node-RED using these instructions (https://nodered.org/docs/getting-started/raspberrypi).

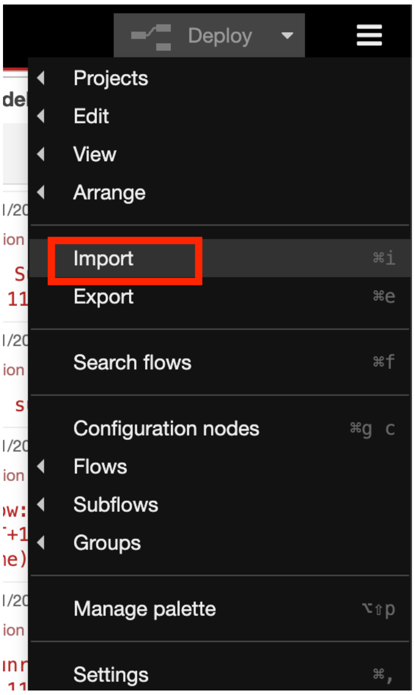
Access your Node-RED dashboard by going to YOUR_PIS_IP:1880

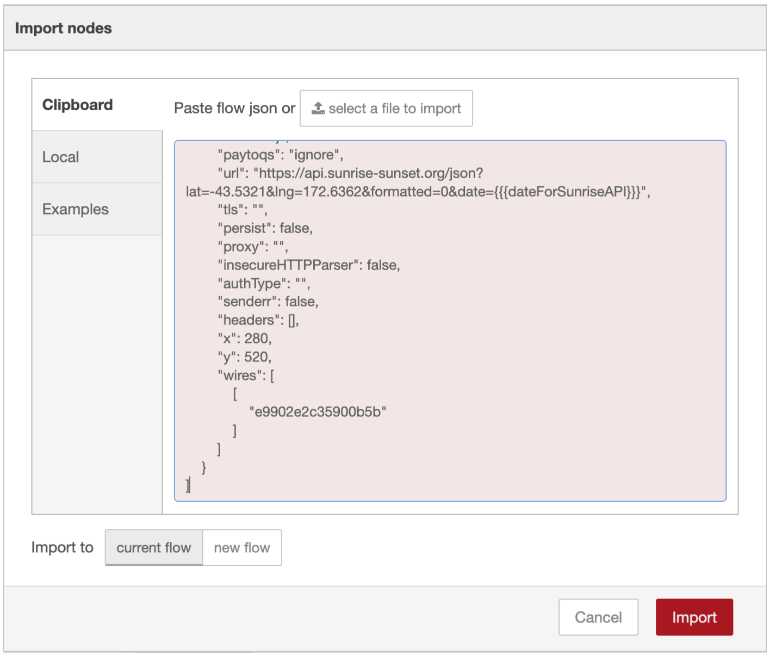
Then upload the JSON code for the flow from here

Access the import section through the hamburger menu.



Press import and copy the Repo code and paste it and press import.





```
        "paytoqs": "ignore",
        "url": "https://api.sunrise-sunset.org/json?
lat=-43.5321&lng=172.6362&formatted=0&date={{{dateForSunriseAPI}}}",
        "tls": "",
        "persist": false,
        "proxy": "",
        "insecureHTTPParser": false,
        "authType": "",
        "senderr": false,
        "headers": [],
        "x": 280,
        "y": 520,
        "wires": [
            [
                "e9902e2c35900b5b"
            ]
        ]
    }
]
```

We will use this file and adjust it to work with your own location and Alexa devices.

Jaxku - Github

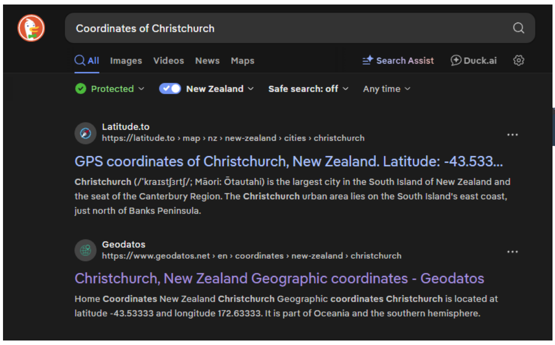
For location, we can use a free API software called Sunrise-Sunset (sunrise-sunset.org)

Sunrise sunset will give a JSON response when the API is called, our Node-RED function can automatically convert that into just the time that the sun will rise and another function will change it to the time-zone of the end-user (which can be adjusted for each user).

How to get the API for your city:

Step 1: Gather the Coordinates for your city

Sunrise-Sunset returns data from Coordinates not from City names so you will need to gather the coordinates for the city you want to operate the device in. This can be done with a simple search.



Coordinates of Christchurch

Latitude.to
https://latitude.to › map › nz › new-zealand › cities › christchurch

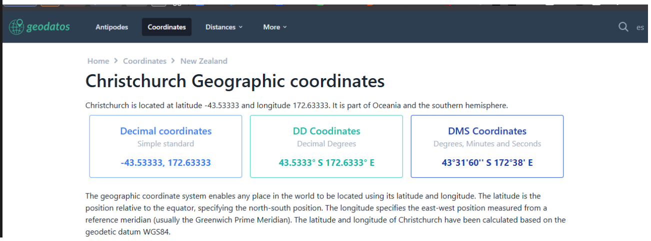
GPS coordinates of Christchurch, New Zealand. Latitude: -43.533...

Christchurch (/ˈkraɪstʃɜrtʃ/; Māori: Ōtautahi) is the largest city in the South Island of New Zealand and the seat of the Canterbury Region. The **Christchurch** urban area lies on the South Island's east coast, just north of Banks Peninsula.

Geodatos
https://www.geodatos.net › en › coordinates › new-zealand › christchurch

Christchurch, New Zealand Geographic coordinates - Geodatos

Home **Coordinates** New Zealand **Christchurch** Geographic **coordinates Christchurch** is located at latitude -43.53333 and longitude 172.63333. It is part of Oceania and the southern hemisphere.



Christchurch Geographic coordinates

Christchurch is located at latitude -43.53333 and longitude 172.63333. It is part of Oceania and the southern hemisphere.

Decimal coordinates Simple standard	DD Coodinates Decimal Degrees	DMS Coordinates Degrees, Minutes and Seconds
-43.53333, 172.63333	43.5333° S 172.6333° E	43°31'60'' S 172°38' E

The geographic coordinate system enables any place in the world to be located using its latitude and longitude. The latitude is the position relative to the equator, specifying the north-south position. The longitude specifies the east-west position measured from a reference meridian (usually the Greenwich Prime Meridian). The latitude and longitude of Christchurch have been calculated based on the geodetic datum WGS84.

From a simple search I can see that Christchurch is located at latitude -43.53333 and longitude 172.63333 which I can now use that information for getting my API request.

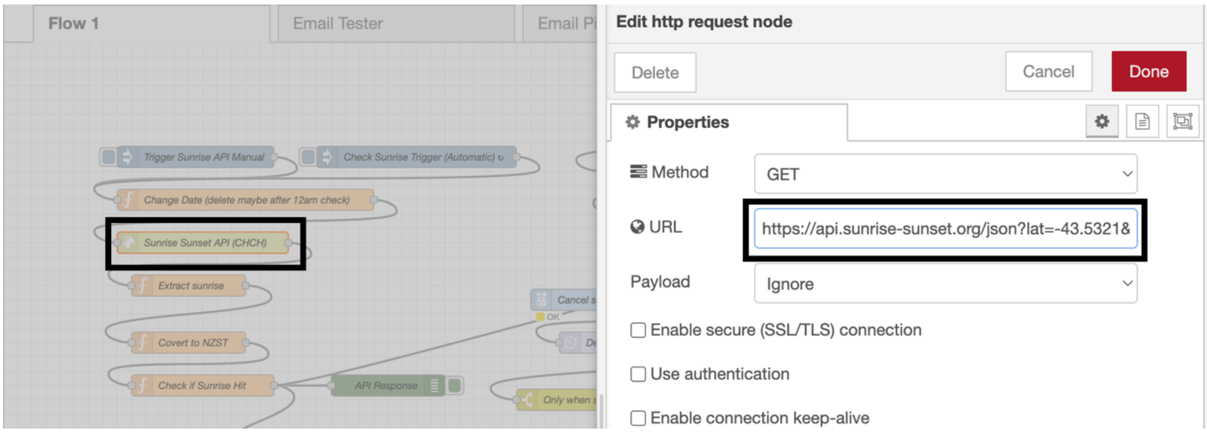
Step 2: Add Coordinates to a request URL:

Now that I have the Coordinates I can add them to a request URL; use this template and fill in the coordinates for the city you want to use.

https://api.sunrise-sunset.org/json?lat=**INSERT_LAT_HERE**&lng=**INSERT_LONG_HERE**&formatted=0&date={{{dateForSunriseAPI}}}

My link for Christchurch after adding in my Coordinates is:

https://api.sunrise-sunset.org/json?lat=-43.5320&lng=172.6306&formatted=0&date={{{dateForSunriseAPI}}}



Method: GET

URL: https://api.sunrise-sunset.org/json?lat=-43.5321&

Payload: Ignore

□ Enable secure (SSL/TLS) connection

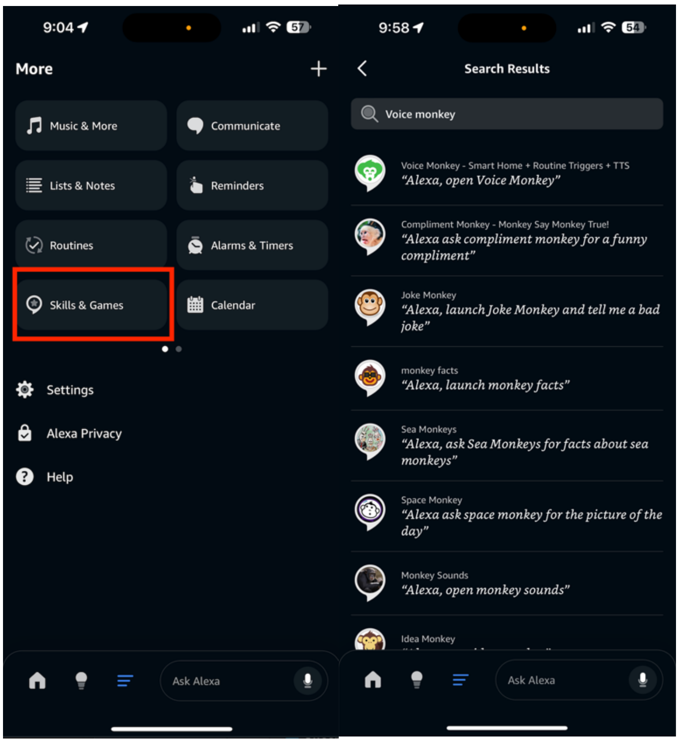
□ Use authentication

□ Enable connection keep-alive

Step 3:

Adjust the Node-RED node to include your API link.

Change the Sunrise Sunset API URL to the one you created


9:04
More
Music & More
Communicate
Lists & Notes
Reminders
Routines
Alarms & Timers
Skills & Games
Calendar
Settings
Alexa Privacy
Help
Ask Alexa
9:58
Search Results
Voice monkey
Voice Monkey - Smart Home + Routine Triggers + TTS
"Alexa, open Voice Monkey"
Compliment Monkey - Monkey Say Monkey True!
"Alexa ask compliment monkey for a funny compliment"
Joke Monkey
"Alexa, launch Joke Monkey and tell me a bad joke"
monkey facts
"Alexa, launch monkey facts"
Sea Monkeys
"Alexa, ask Sea Monkeys for facts about sea monkeys"
Space Monkey
"Alexa ask space monkey for the picture of the day"
Monkey Sounds
"Alexa, open monkey sounds"
Idea Monkey
Ask Alexa

Step 4:

Setting up voice monkey (This will tie into your Amazon Account)

Enabling the skill

1. Open the Alexa app on your phone
2. Go to More -> Skills & Games
3. Search for “Voice Monkey”
4. Tap enable to use
5. Then follow the prompts to link Voice Monkey with your Amazon account.

Step 5:

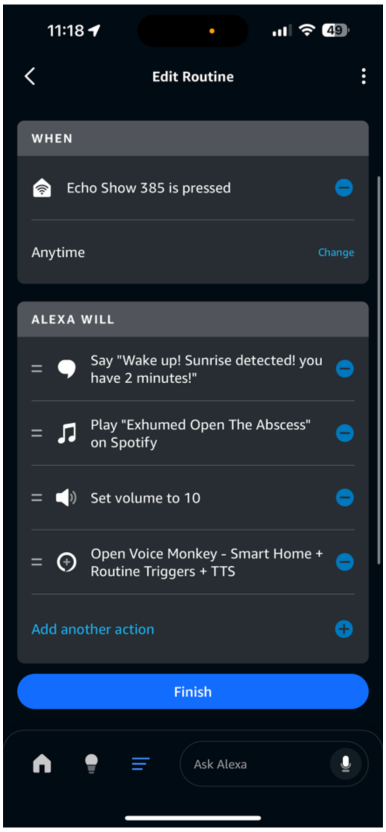
Creating the voice monkey account online:

1. Go to voicemonkey.io
2. Click sign up / Login with Amazon
3. Sign in with the same Amazon account linked to your Alexa
4. Complete the linking process on the website and creating the routine as instructed on the Echo app, remember the device name you give the API device make it unique this is for your starting call.
5. Make another device and follow the same steps this will be for your stop call.

Step 6:

Creating the wakeup Amazon Routine:

1. More -> Routines (+).
2. Set 'when' to smart home -> When 'your API device name start' is pressed.
3. Alexa will: Say 'Wake up! Sunrise detected! You have 2 minutes!'
4. Turn on lights (Optional but make sure your IoT lights are connected to your Alexa account.
5. Set volume to 10
6. Play 'Exhumed Open The Abscess' or any or any other loud song of your choice. (Depends on your music streaming service and subscription status)



Edit Routine

WHEN

Echo Show 385 is pressed

Anytime Change

ALEXA WILL

Say "Wake up! Sunrise detected! you have 2 minutes!"

Play "Exhumed Open The Abscess" on Spotify

Set volume to 10

Open Voice Monkey - Smart Home + Routine Triggers + TTS

Step 7:

Creating the stop Amazon Routine:

1. More -> Routines (+).
2. Set 'when' to smart home -> When 'your API device name stop is pressed.
3. Add stop audio
4. Set volume to 3 (Or whatever volume you'd like your Alexa to be normally at)

Step 8:

Getting your API Credentials.

1. Click on API V2
2. Click 'Announcement API' and copy the API URL
3. In Node-RED:
   Change the URL in the 'Trigger Alexa via 'Voicemonkey' node to your personal start API URL.
   Do the same but for the stop API URL in the 'Stop Alexa via VoiceMonkey'

Step 9:

Setting up email.

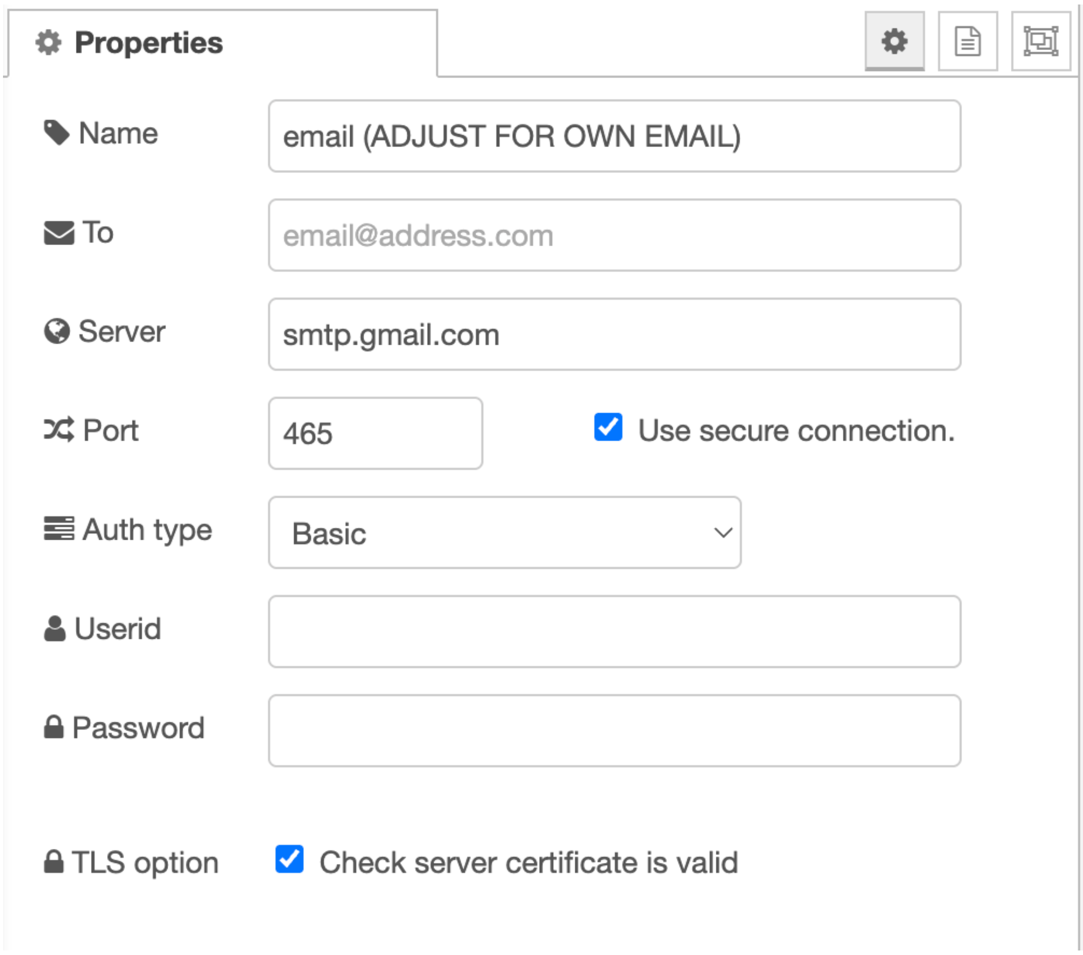
This setup process will be different email providers the simplest provider is using Gmail for this to work you need to get an app password to use within Node-RED (Not your actual Google password), if you use a different Email provider you'll need their email server address and the port they use, you will also need to find out how to generate an App password for your account (A search engine will be your best friend here)

Follow this guide here to get your Google app password (Note you need 2FA on your account to be able to generate a app password). I would also recommend deleting your app password when it is no longer needed or skipping this step altogether if you don't want an email notification and want higher security.

Your Userid will be your Gmail email/username (e.g. CoolGuy@gmail.com)

The password will be the App password **NOT** your Google account password.

In Node-RED:



Properties

Name: email (ADJUST FOR OWN EMAIL)

To: email@address.com

Server: smtp.gmail.com

Port: 465 ☑ Use secure connection.

Auth type: Basic

Userid:

Password:

TLS option ☑ Check server certificate is valid

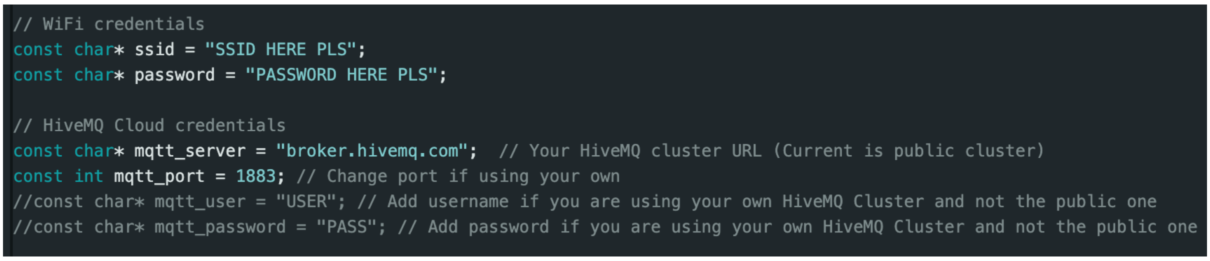
Step 10:

Adjusting other nodes.

Other nodes may need to be adjusted, e.g. if you are operating this out of New Zealand nodes like the ‘Convert to NZST’, and ‘Change Date’ as the time zones will need to be changed, GenAI can help you with this if you don’t have JavaScript knowledge.

Step 11:

HiveMQ Setup:



```
// WiFi credentials
const char* ssid = "SSID HERE PLS";
const char* password = "PASSWORD HERE PLS";

// HiveMQ Cloud credentials
const char* mqtt_server = "broker.hivemq.com";  // Your HiveMQ cluster URL (Current is public cluster)
const int mqtt_port = 1883; // Change port if using your own
//const char* mqtt_user = "USER"; // Add username if you are using your own HiveMQ Cluster and not the public one
//const char* mqtt_password = "PASS"; // Add password if you are using your own HiveMQ Cluster and not the public one
```

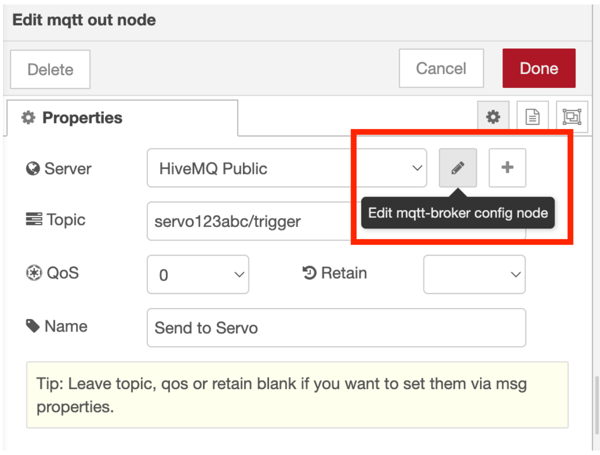
In the SweepWIthHIVEMQ.ino file open with Arduino IDE and select the board (e.g. D1 Mini)

In the code spots for you to adjust are clearly marked, such as the SSID and the password.

To connect to HiveMQ you can either create an account on the service and create your own private cluster (if so you will need to put in your MQTT user and password with its server link and port (*Remember to uncomment the user and pass) or use the default public cluster that is already listed in the code and give yourself a unique topic name (remember this as it is your 'password' and is needed within the Node-RED flow.)

Note: Using the public cluster comes at a risk. It is public the only protection you have over it is the topic name, therefore if someone knows or uses the same topic name as you they can trigger the servo.

Edit the send to servo node with your information and upload the updated code to your Wemos board.



Edit mqtt out node

Server: HiveMQ Public

Topic: servo123abc/trigger

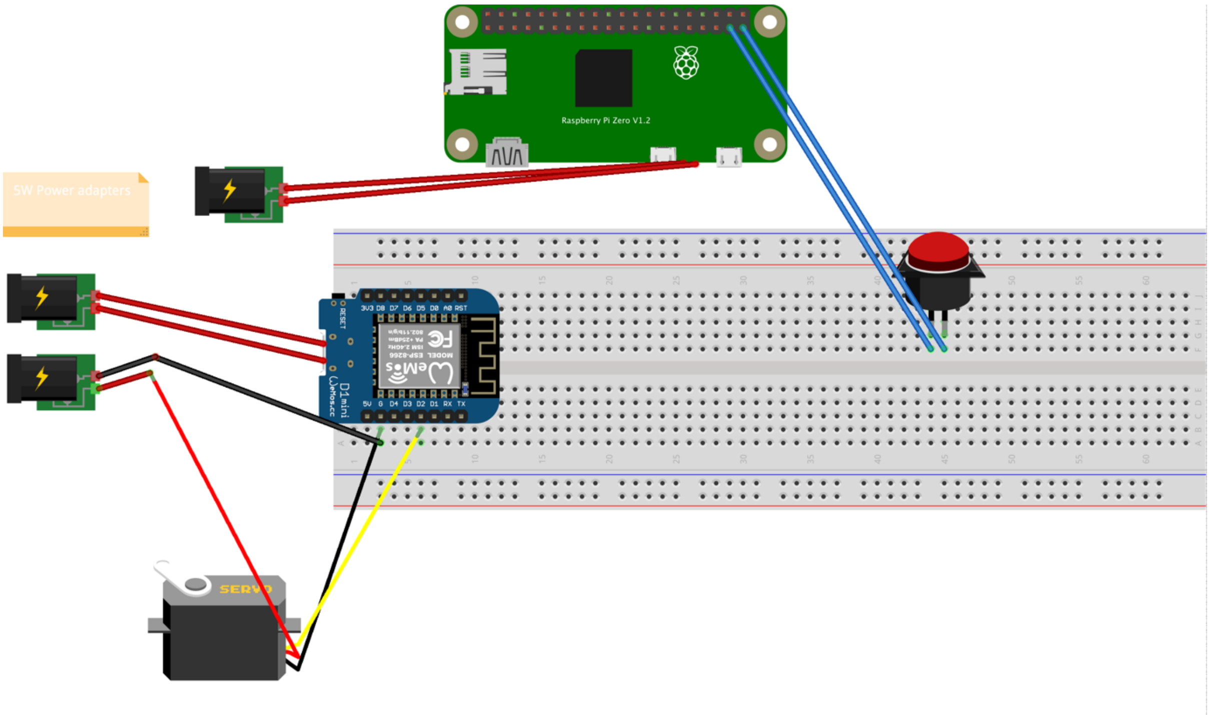
QoS: 0

Retain:

Name: Send to Servo

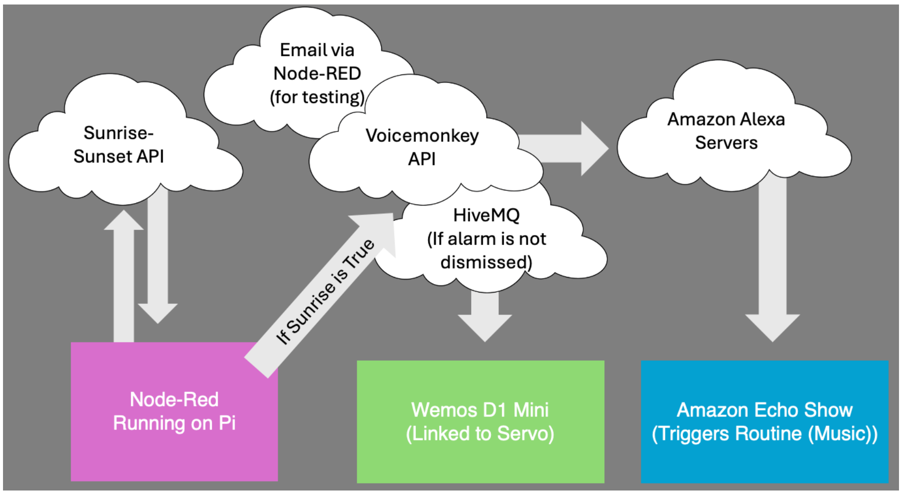
Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

Technical details:



Wiring diagram, all powered using 5w power adapters (such as a standard Apple iPhone charger) or battery banks.

Cloud diagram on how the Wemos can communicate with Node-RED on the Pi:


Email via Node-RED (for testing)
Sunrise-Sunset API
Voicemonkey API
Amazon Alexa Servers
HiveMQ (If alarm is not dismissed)
If Sunrise is True
Node-Red Running on Pi
Wemos D1 Mini (Linked to Servo)
Amazon Echo Show (Triggers Routine (Music))