**Campernet**

**A whole new type of internet.**

****

**By Andy Lear**

Contents

[Introduction 3](#_Toc208852450)

[Microcontroller based project ideas 3](#_Toc208852451)

[Weather station 3](#_Toc208852452)

[Doorbell/security system 3](#_Toc208852453)

[Caravan/catch all 3](#_Toc208852454)

[Other tech project ideas 3](#_Toc208852455)

[Media server 3](#_Toc208852456)

[Data server 4](#_Toc208852457)

[Our own internet. 4](#_Toc208852458)

[A host of advantages. 4](#_Toc208852459)

[Disadvantages 4](#_Toc208852460)

[The Campernet Lanifesto 4](#_Toc208852461)

[Structure 5](#_Toc208852462)

[Figure1. Domain structure for a single user. 5](#_Toc208852463)

[Further requirements 5](#_Toc208852464)

[An ‘owner’s tier’ 5](#_Toc208852465)

[Figure2. Domain structure for multiple users. 6](#_Toc208852466)

[Security and authentication 6](#_Toc208852467)

[An acceptable risk? 6](#_Toc208852468)

[A secure browser 6](#_Toc208852469)

[The campernet search engine 6](#_Toc208852470)

[Keeping up with the times. 6](#_Toc208852471)

[Voice assistant. 7](#_Toc208852472)

[LLM (Large Language Model) 7](#_Toc208852473)

[Conclusion 7](#_Toc208852474)

## Introduction

I have three main microcontroller based project ideas for projects for when I am camping or caravanning, a weather station and a doorbell/security system and a sort of catch all which I will call caravan stuff although some of it will apply to camping as well.

## Microcontroller based project ideas

### Weather station

This will have based around the ISP32 microcontroller and the BME280 sensor which measures temperature, humidity and barometric pressure. To this I would like to add a tipping bucket rain gauge and wind speed and direction sensors. These are the basics although there would be the option to add ‘nice to have’ sensors like solar radiation, air quality etc. You would like to think that with enough data some machine learning could make a better job of local weather forecasting than the Met office as well.

### Doorbell/security system

Rather than building this from the ground up I have bought a Byron portable wireless doorbell set which simply consists of a button and a chime. This is because I would not be able to make a decent button enclosure. The doorbell plan would be to hack into the chime and connect that to an ISP32 via an opto-isolator. The ISP32 could then notify my phone via wireless or Bluetooth if I was out of the caravan but near. If I was not near then SMS messaging would be an option that I don’t like because it involves SIMs and subscriptions. Meshtastic would be a preferable option. The obvious security additions would be a camera and PIR (passive infrared) sensors, radar or whatever. It might be fun to have a servo operated water cannon or nerf gun mounted on top of the caravan. I am getting a little silly here but a fog machine under the door would make for a dramatic entrance.

### Caravan/catch all

I recently learnt that it is possible to measure (roughly) the amount of gas left in a cylinder with a JSN-SR04T ultra sonic sensor. This would be a great asset. It would be lovely to measure the battery power left as well but I am not going to lie, this would be a total ball ache for the Jackery because although I can see how much it has on the app this is a propriety system. It would be theoretically possible to connect a camera to an esp32 and use some sort of OCR system to read it. It might be possible to use that as an intermediate step and by sniffing the Bluetooth packets and comparing them with the readings on the jackery decode the packets but it would be a total nightmare if the jackery system uses rotating encryption which it almost certainly does. It would not be so hard to measure the charge left on the power banks that would power some of the other ISP32s. It would be lovely and quite easy to have some way to find my way back to the caravan and no doubt there are many other possible additions to the catch all system.

## Other tech project ideas

Having started to go down the rabbit hole of how to enhance camping and caravanning with tech several other possibilities present themselves that could be done with microcontrollers but could as easily (probably more easily) be done on a laptop.

### Media server

There may well be times when I want to listen to music. I have some of my favourite music on a USB stick that I play in the car. Obviously I could just plug the stick into a laptop and if it had some decent external speakers play it like that. Laptops have tons of storage and even if they didn’t it would be easy enough to plug in an external drive. It would be much harder to control what the computer was playing remotely from my phone. We are then looking at some sort of media server that could also host books, films and possibly even games.

Data server   
So you are off grid arguing with a friend about who sung ‘I’m a Barbie girl” or weather a spider has six or eight legs. Using a piece of software called zimit you can download entire websites as .zim files. These can be read with another piece of software called kiwix (which will run on pretty much anything, including raspberry pies). Using .zim files the entire English version of Wikipedia (including images) would take less than 150 Gb.

## Our own internet.

You could download any website for offline use. Add to this all the other items we have mentioned and you really do have a great deal of data. Obviously it is a drop in the ocean compared to the normal internet but if we were to add more users, all with their own mini servers (which is effectively what these ESP32 chips are) then you would have something pretty close to internet access with no internet. What else does the internet have? Email and/or messaging. Search engines and browsers, social media and there is no reason for Campernet not to have all of these as well.

A host of advantages.   
I could not resist the pun. Starting an internet from scratch could have many advantages. The main one is that it would work off grid. Secondly it would not be riddled with ads, tracking cookies, paywalls and everyone wanting to make money out of it. If some ads snuck in in via downloaded websites there is an option called pi-hole, a network wide raspberry pie based ad blocker. Thirdly it would be out of the reach of governments and the tech companies that own the internet infrastructure. Fourthly it could grow into a genuine community and lastly it could be fun.

### Disadvantages

The obvious disadvantages are that it would take time, money and effort. However, since it would start small and grow at its own pace these resources do not need to be found up front. Apart from the host of technical challenges there will also be the temptation for greed or convenience to creep in making Campernet just a niche alternative to the internet we have at the moment. To guard against such creep I propose a sort of Campernet constitution. Since it will start in the UK this might better be referred to as the Campernet lanifesto.

The Campernet Lanifesto.

**1 There is no such thing as a necessary cookie.**  
 All cookies are unnecessary and most are unwanted. The only exception is those that remember passwords — and this task belongs to the Campernet browser, which stores them locally, securely, and without drama. Ads, trackers, and surveillance are left at the gate.

**2 Every update must be uninstallable.**  
 Developers are imperfect and mistakes will happen. Rollback is not a luxury — it is a necessity.

**3. Offline is not broken.**  
 It is intentional. Campernet thrives without WAN. No “you’re not connected” errors — just making do with what resources you have.

**4. Every module must earn its place.**  
 No fluff, no bloat. If it doesn’t serve a purpose, tell a story or make people smile then it doesn’t belong.

Structure

The obvious place to start is with a domain and the obvious name for that would be .campernet. We have already identified a number of components of campernet. The following figure is a radically simplified chart that does not have the social and many other subdomains and sub sub domains etc. It would give the general concept if there was only one user of campernet, which clearly there will not be.

## **Figure1. Domain structure for a single user**.

.charge\_left

.charge\_rate

.campernet

.caravan

.velocity

.volume

.music

.power

.gas

.gutenberg

.wikipedia

.rain

.films

.wind

.media

.data

.weather

### Further requirements

This is obviously not going to work for more than one user because if there were two users and each had a gas sensor then they would obviously be far more interested in how much gas they had left than how much gas the other person had and there would need to be a way to distinguish between the two. As much to the point, if for example Andy had a data server with Wikipedia on and Fred had one containing Project Gutenberg, Andy could read Fred’s data (if he had permission) but when Fred went home then as far as Andy is concerned, the data would disappear. This could be addressed with either a custom device discovery broadcast but I believe that the current ESP-MESH framework and protocol might be able to cope with it.

### An ‘owner’s tier’

It is also becoming apparent that there will need to be an extra tier inserted into the domain structure. The top level .campernet domain is never going to change. Nor is the next level down (the second level) which can be hard coded because it amounts to a list of categories. In potential defiance (if there were a lot of users) of the ‘rule of thumb’ that domains should start with the most general and get progressively more granular I believe the right place to put the owners tier is directly under the second level. The rational here is that when a user went home then all their owned data categories as well as the data itself would disappear from the tree, making it more manageable for discovery, indexing and searching.

Figure2. Domain structure for multiple users.

.campernet

.power

.charge\_rate

.charge\_left

.data

.caravan

.power

.gutenberg

.andy

.charge\_left

.charge\_rate

.caravan

.data

.wikipedia

.fred

## **Security and authentication**

### An acceptable risk?

This opens up all sorts of possibilities when it comes to security and authentication. It would be eminently possible to log on to the system once and have (or grant or revoke) access to everything for which you have the necessary permissions. This might be a double edged sword because while this would make everything more convenient it would also mean that if the account was compromised then the miscreant would have access to everything. That said, there should be nothing of any great financial value on campernet, it will not ask for credit cards or bank details so the consequences would not be as bad as it would be on the internet.

### A secure browser

Security could be enhanced by only allowing access to campernet via a specific browser which we would probably need anyway to deal with the campernet domain (in the same way as the dark web uses its Tor (The Onion Router) Browser to browse the onion domain.

### The campernet search engine

Another thing that will be required is our own search engine because google won’t work offline and even if it did with all its data gathering, targeted add serving, and general security misbehaviour we would not want to use it anyway. A new browser (probably based around firefox) will be required and the name Campoogle suggests itself.

### Keeping up with the times.

Whilst a secure browser will undeniably be necessary it is a bit last century and nowadays google and chrome are being supplanted by Alexa and Siri. We are also undeniably in the age of artificial intelligence (without which this document and indeed the entire campernet project) would not exist. Any voice assistant worth talking to would need access to an LLM (large language model) such as ChatGBT, Microsoft Co-pilot or Google Gemini. However one step at a time.

### Voice assistant.

There is a downloadable voice assistant called Mycroft (named after Sherlock Holmes’ older and smarter brother). There is also a lightweight version called Picroft that runs on raspberry pie OS. What it can do without an LLM back end is tell you what the weather is, what the time and date is, answer questions based on Wikipedia and play music. However it expects access to the internet to do this. Apparently with a bit of friggary it can use the stuff we have been discussing. The two approaches would be either to write a custom picroft script to interrogate the weather station or have the weather station ‘publish’ an MQTT topic like weather/temperature to (and using) a broker called mosquito which is tiny and could live on the weather station pi. For general knowledge questions you would need to write a Mycroft skill that queries Kiwix via its HTTP API. It could play music from the media server but this would probably be trickier because the music will not always be in the same place but as long as the media server(s) is/are running a Music Player Daemon it is possible, as would be writing a picrofr script to tell you how much gas was left in the gas cylinder etc.

LLM (Large Language Model)  
Also known as a neural net or artificial intelligence. This would be the icing on the cake. There are lightweight LLMs the can just about be run on a raspberry pie but it is best to run the LLM on either a decent spec laptop or a single board computer such as the Minisforum UM890 Pro running Linux. The best LLM I can find is the Dolphin LLaMA3 uncensored. The reason this is so good is that it is as free from other people telling you what you can or can’t know as possible.

## Conclusion

This is the bare bones of an idea. I have barely touched on messaging and email other than mentioning meshtastic which would only be a part of it. Because people will have their own music the media server will be not much short of virtual, and if you want to see what someone else is listening to, or has listened to, then you are pretty much going to have to build your own Spotify. There will be mistakes and omissions in this document which will hopefully be corrected as more stuff is learnt. Hopefully at the very least it is food for thought.