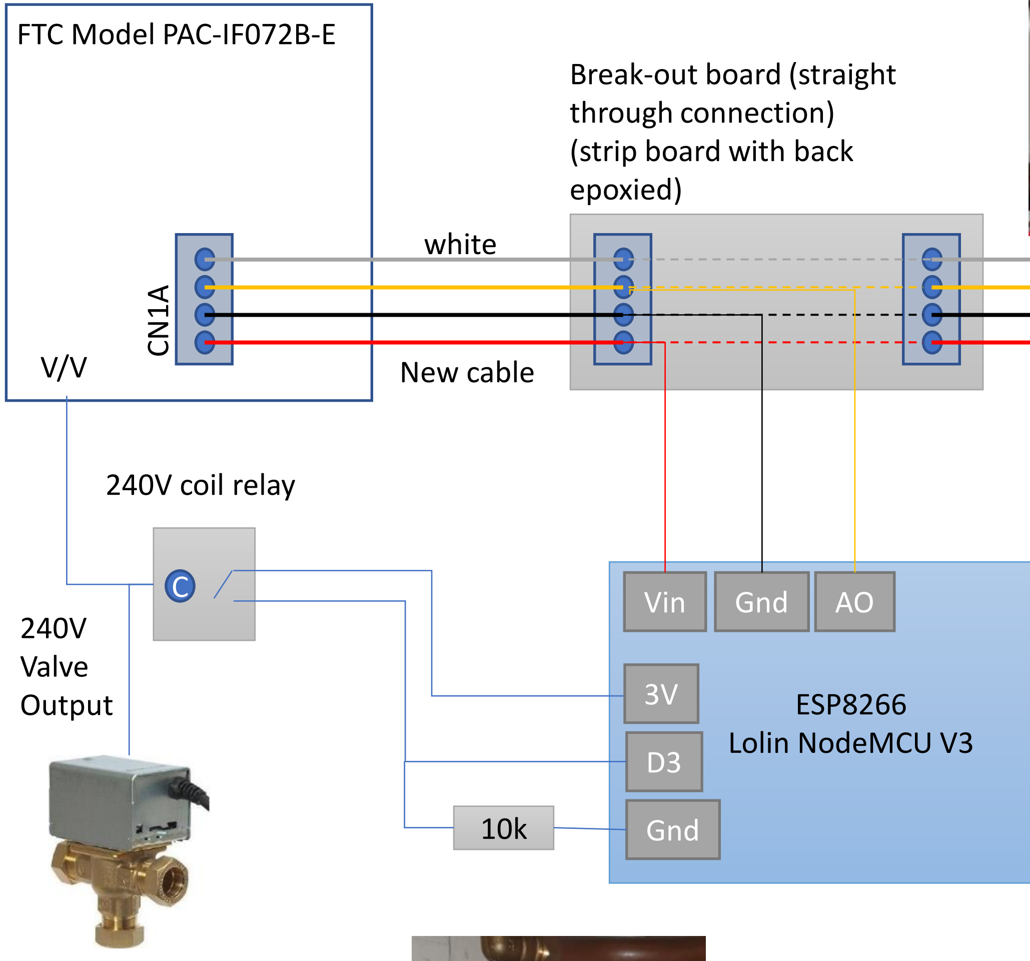
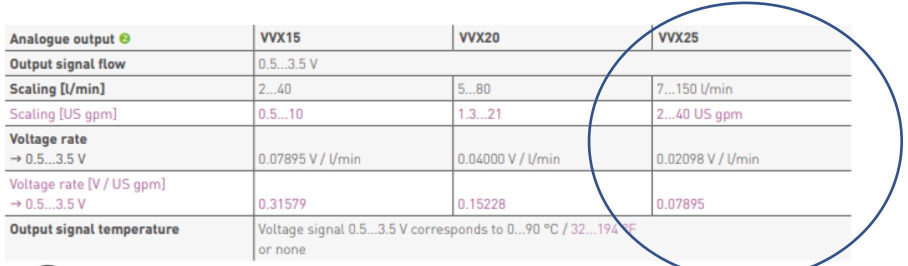
Piggy-backing an ESP8266 onto an Ecodan Sika Flow Meter

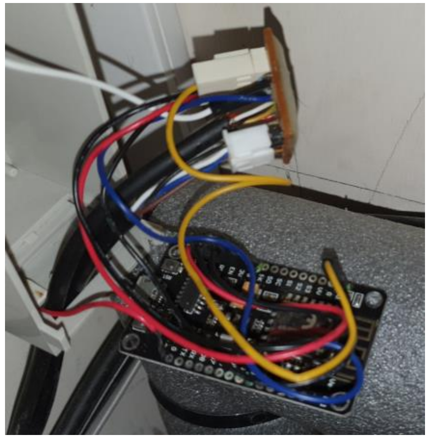


Original cable



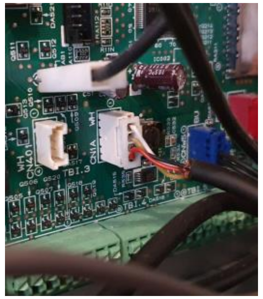
Analogue output	VVX15	VVX20	VVX25
Output signal flow	0.5…3.5 V		
Scaling [l/min]	2…40	5…80	7…150 l/min
Scaling [US gpm]	0.5…10	1.3…21	2…40 US gpm
Voltage rate → 0.5…3.5 V	0.07895 V / l/min	0.04000 V / l/min	0.02098 V / l/min
Voltage rate [V / US gpm] → 0.5…3.5 V	0.31579	0.15228	0.07895
Output signal temperature	Voltage signal 0.5…3.5 V corresponds to 0…90 °C / 32…194 °F or none		

My flow meter is a VVX25,
I concluded that the flow would never be high enough to reach the full 3.5V output – the ESP8266 only reads up to 3V, but the input could handle 3.5V.



These are the connectors I used to go onto the FTC board – they aren't perfect, but they were the only ones I could find that would fit!
https://www.ebay.co.uk/itm/232392254374

I've not shown the terminals for the valve/relay because I think they will be different for other peoples installs depending on configuration. I had the relay lying around, so used it, there may be better ways to do this.



What it looks like all boxed up!

I've included the sketch for the Arduino as a .txt file. It is probably terrible – it's the first time I've ever written an Arduino Sketch and mostly just found other bits to cut and past ☺

It writes the data direct to the web-based EMONCMS, not to your local EmonPi.

I did try MQTTing it, but struggled to get that to work and as I wanted to log and use it in EMONCMS this works fine for me.