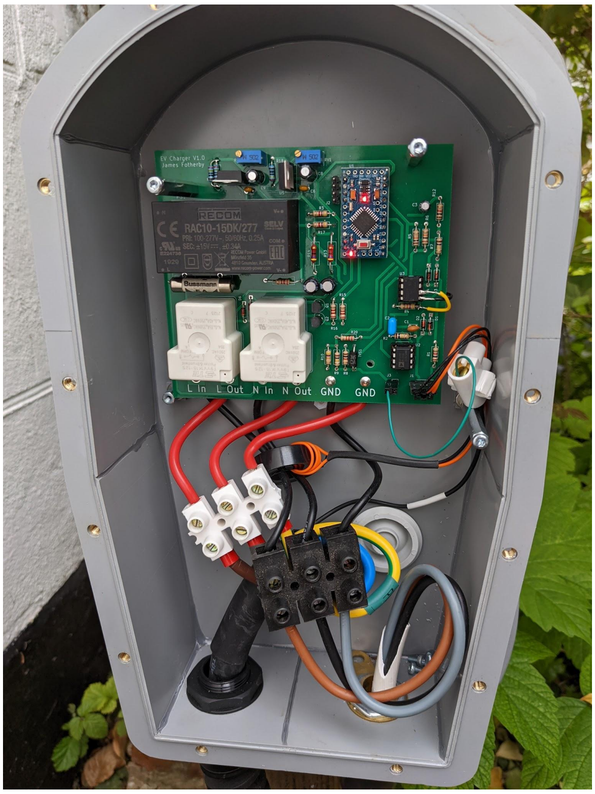
EV Charger

James Fotherby, March 2022

Abstract

I built my own 7.2kW EV charger and fitted it inside a Zappi enclosure. I also wrote the Arduino software for it. The 2 aims were simplicity and safety. This article documents the build. All design files, software and part lists are available on the GitHub page.



Motivation and Disclaimer

It costs about £900 in the UK to have an EV charger installed by an electrician. This motivated me to question whether I could build my own. On researching the topic further I found an open source charger (the EVSE) with good documentation. This gave me the confidence to build my own.

A word about safety. I'm not an electrician. I'm not allowed to make alterations to my consumer unit. I will have to remove the charger when I leave my house. I'm sure there are countless people that would condemn what I have done here and I can understand why. Building your own outdoor charger which switches 240V 32A could be dangerous if not done right. I have learnt about earthing systems, PEN faults, RCDs, cable current capacities etc. I think I have educated myself enough to have built an adequately safe system. Nevertheless, I welcome constructive criticism and discussion.

Introduction

EV chargers use a simple “pilot” signal to detect when they are plugged into a car and to tell the car how much current it is allowed to draw from the charger. They don’t modify the mains at all, they just switch it on/off to the car via some relays. In addition to this they also incorporate the functionality of an RCD. But to be honest, that’s about it!

I managed to buy a second hand “showroom” Zappi charger. It came without any of the electronics inside but it gave me an enclosure, cable and plug to work with. I paid £120 inclusive of postage.

I bought 5 metres of 6mm² SWA cable to run from my consumer unit to my desired charger location. I added a 50A MCB into my consumer unit on the non-RCD side and routed the SWA cable using cleats and stainless steel screws.

The SWA cable enters the Zappi enclosure through an outdoor water resistant gland. The live and neutral pass together through a current transformer before attaching to the PCB.

Ground Current Detection

One of the most important safety mechanisms to include is a ground current detection system. The chassis of the car is grounded via the earth wire through the charging plug. The earth supply comes from the consumer unit (we have TN-C-S supply).

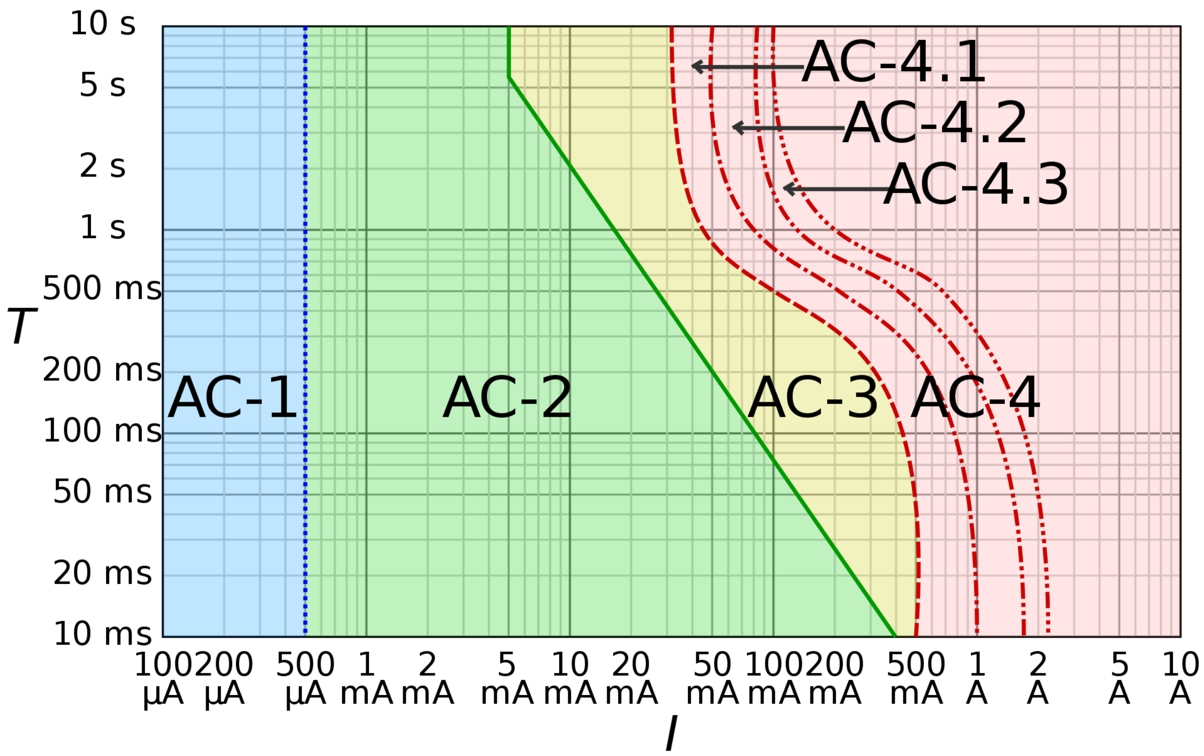
There is quite a bit of theory behind grounding systems. John Ward has some instructional YouTube videos on the topic which I have watched. He discusses the problem of PEN faults etc. It’s worth spending the time educating yourself about earthing if you are doing any electrical work.

Although unlikely, it is possible for a fault to occur such that a live wire contacts the car chassis. Perhaps it's pulled loose inside the car somewhere and is touching the chassis or perhaps a wet connector is bridging a path to the chassis.

Either way the live will source current into the chassis which will drain straight to ground (In a TN-C-S supply the ground and neutral conductors are bonded at the consumer unit). The amount of current will depend on the resistance of the faulty bridge. (water is unlikely to allow many amps to flow). Given the chassis is well earthed, it shouldn’t rise up in voltage enough to present a shock hazard to someone that touches it.

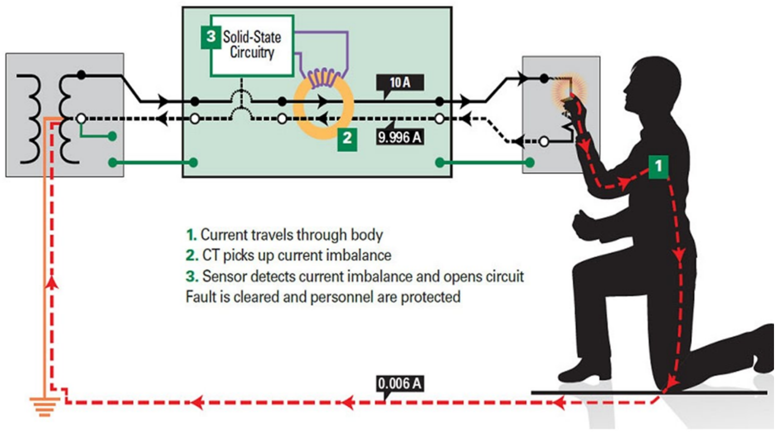
Nevertheless, this is a fault situation that should be detected and dealt with. If some water is bridging the live to the chassis perhaps a couple of amps will flow (not enough to trip the 50A MCB for the charger) but enough to cause localised heating and further damage.

So we need to measure current flowing to ground (should be zero in normal operation). If this is more than 20 mA we want to isolate the car by opening the relays. The reason RCDs typically trip at 5-30mA is because this amount of current for a couple hundred milliseconds doesn't cause permanent injury to humans. I like the [wikipedia article on electrical injury](https://en.wikipedia.org/wiki/Electrical_injury).



AC-1: imperceptible, AC-2: perceptible but no muscle reaction, AC-3: muscle contraction with reversible effects, AC-4: possible irreversible effects, AC-4.1: up to 5% probability of ventricular fibrillation, AC-4.2: 5–50% probability of fibrillation, AC-4.3: over 50% probability of fibrillation

The way to measure current to earth is simple. We employ a current transformer measuring the common mode current of the live and neutral wires. All current should be differential (all current flowing out of the live wire should go through the load and return back through the neutral). If there is a fault and some current does not return then it must be going to earth. This is a common mode current and we want to be measuring this!

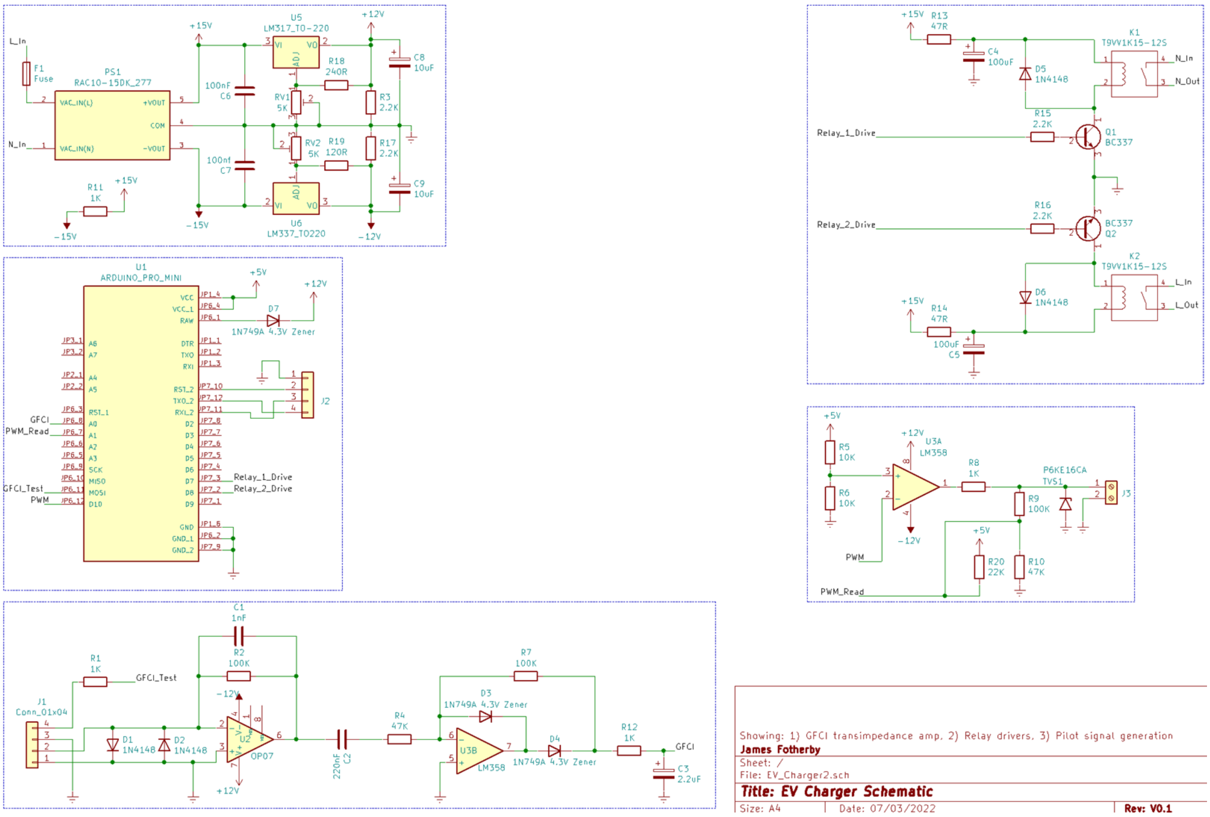


1. Current travels through body
2. CT picks up current imbalance
3. Sensor detects current imbalance and opens circuit
Fault is cleared and personnel are protected

Schematic

The electronics are necessary for:

- Generating the DC voltages for the arduino, op-amps, relays etc
- Mounting of the 40A 250V L & N relays that switch power to the charging plug
- Generating the +/- 12V 1kHz PWM signal for the pilot
- Amplifying and rectifying the current transformer's signal prior to the Arduino's ADC



Power Supplies

I used a RAC10-15DK/277 AC/DC power module. This generates +/- 15 V rails. Adjustable positive/negative linear voltage regulators (LM317 & LM337) produce +/- 12V rails. I was aware the op-amps outputs might not be able to swing all the way to their supply rails so I wanted some flexibility by using adjustable voltage regulators.

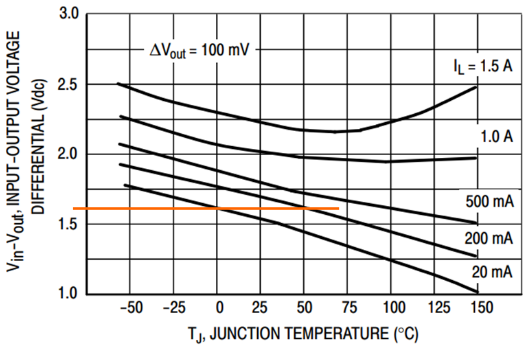


Figure 10. Dropout Voltage

The regulators need a minimum load of about 5 mA to maintain regulation. Therefore R3 & R17 provide a small load for them. The regulators are operating uncomfortably close to their dropout voltages. According to the datasheet the dropout at 20 mA load is about 1.6 V at 0℃ which allows us to up our op-amp rails to about 13.4 V if necessary.

Due to the chip/stock shortages at present, I bought a Pro-Mini module which conveniently housed an Atmega328P Arduino with a 5V regulator. Beware though, this onboard regulator has a max input voltage of 10V so I dropped the regulated 12V using a 4.3V zener diode before supplying it to the RAW input of the Pro-Mini.

Pilot Generation

All communication with the car is done through a single wire referenced to earth called the pilot signal. Read here and here for descriptions on how this signal works. In a nutshell, depending on whether the car is connected/ready to charge etc, the car places different resistances on the pilot signal. This causes changes in the voltage of the pilot signal.

An LM358 Op-Amp takes a 0-5V PWM signal from the Arduino and converts it into a +/-12V signal to form the pilot. Easy.

We use a voltage divider network to condition the pilot voltage before feeding it into an ADC channel for measurement. A 13.6 V 600W bidirectional TVS just ensures no whacky voltages are ever experienced on the pilot wire.

Relay operation

I initially thought I ought to share the load on the SMPS between its 2 power rails. So one relay would be powered by the +ve rail and the other by the -ve rail. However doing this added a couple of extra parts to the design and slightly increased the overall complexity. For the sake of simplicity I kept both relays powered from the +ve supply rail and this has ended up working absolutely fine.

The T9VV1K15-12S relay specifications report a coil holding voltage of only 4.7V. This can conserve a lot of power. You can see from the schematic, we charge 100uF capacitors from the +15V rail through 1W 47R resistors (R13 & R14). When the relays are activated they initially but briefly get 15V. But the steady state voltage decays to about 9V. I should have gone for 68R or even 100R resistors for even more power conservation.

The BC337 transistors get about 2 mA of base current through the 2.2K resistors. This is adequate to sufficiently switch the transistors.

Current transformer

A current transformer is analogous to a voltage transformer. We put 1 turn on the transformer primary by passing the live and neutral once through the ferrite core. On the secondary we have many turns - 100s of turns. A current on the primary is therefore induced onto the secondary albeit at a magnitude proportional to the turns ratio. If we have a 20mA current on the primary and a 1:400 turns ratio we'll have a 50uA secondary current.

Just like a voltage transformer doesn't like to be short circuited. A current transformer doesn't like to be open circuited. The best method to measure the secondary current is to use a transimpedance amplifier.

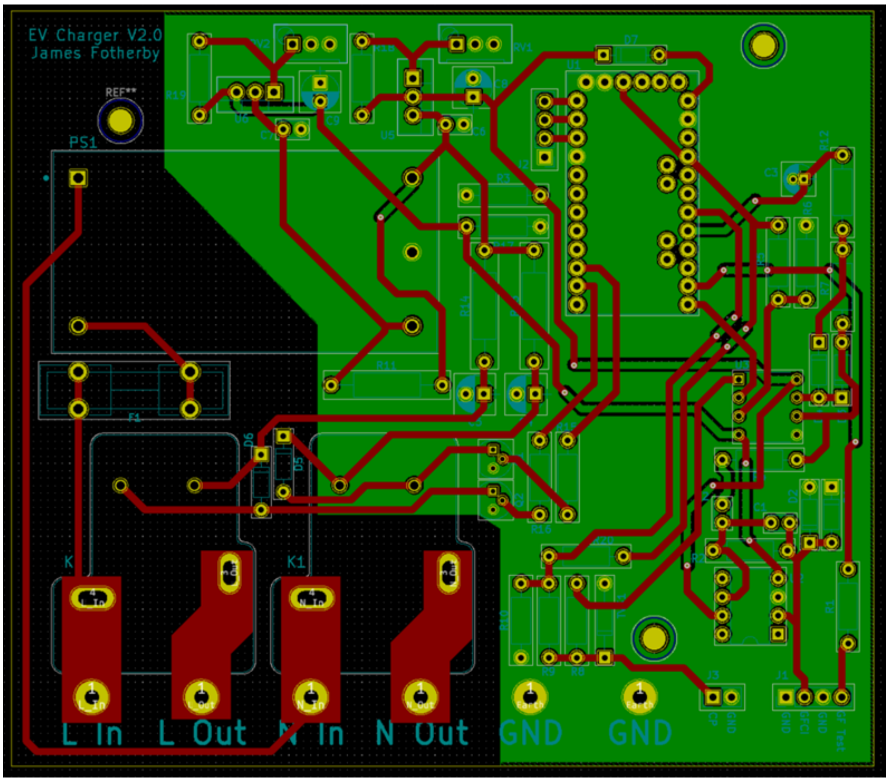
U2 is the OP07 low-offset op-amp. The V+ terminal is grounded and in this configuration the output will swing around to always keep V- at the same voltage as V+ (ie. 0 V). Imagine if the current transformer forces 50 uA towards the V- terminal of U2. The op-amp will reduce its output voltage to -5 V so that the 50 uA is entirely pulled through R2 (V = I x R = 50uA x 100K). Hence the V- terminal is kept at exactly 0 V. So you can see in this configuration that currents from the transformer are converted into voltages at the output of the op-amp. C1 just helps to reduce the gain at high frequencies and act as a low pass filter. D1 and D2 stop any voltage excursions beyond about 0.7 V should the transimpedance amplifier get saturated.

We probably didn't need the low-offset features of the OP07. C2 removes any DC bias anyway. U3B is configured to act as a further amplifier stage and precision rectifier. The 4.3 V zener diodes used clip the output to a ~4 V maximum as well. R12 and C3 add a final low pass filter before going to the ADC channel.

All in all, AC currents from the ground fault current transformer are converted into voltages, amplified, rectified, limited and filtered before being passed to the Arduino ADC for measurement. This circuitry worked and made sense to me so I went with it. But you could probably simplify it further.

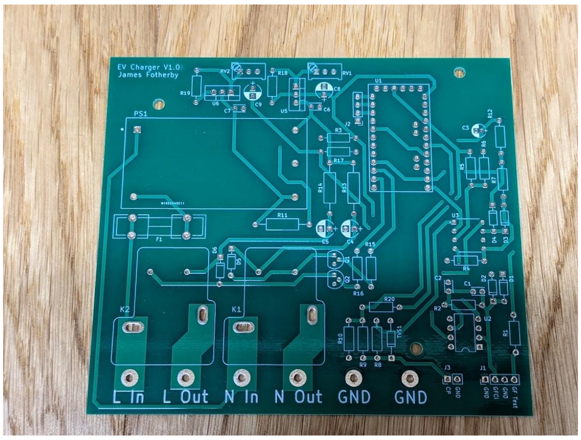
PCB Design

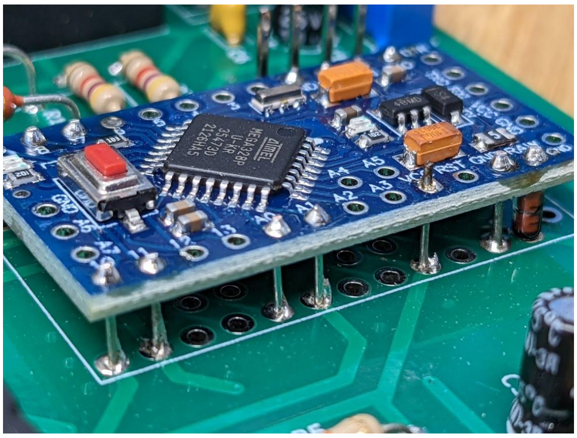
I used KiCad to design the PCB. I have settled on KiCad because it is open source and you can do multi-layer PCBs if you need to. SnapEDA is key for importing PCB footprints for parts. There is not much to say about the PCB design. I kept a 4mm clearance around high voltage traces. I ordered the PCBs from PCBway who get them back to you in about 1 week!

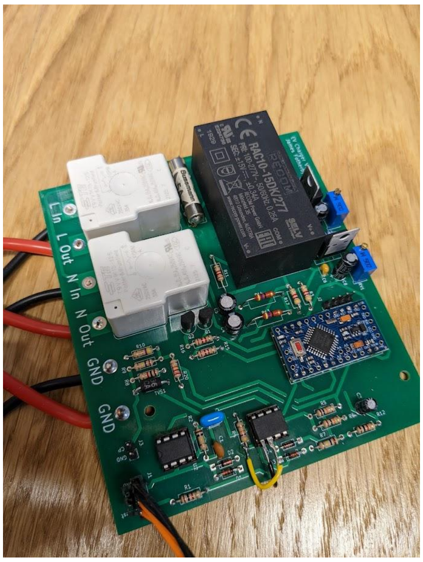


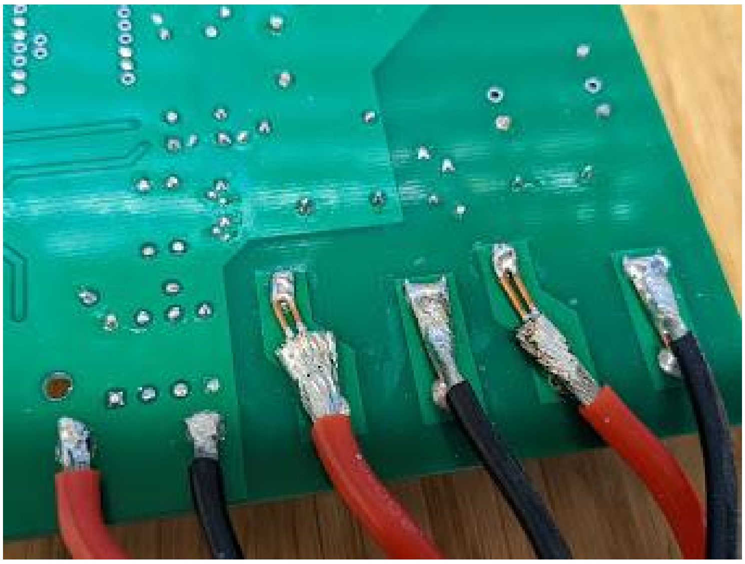
Assembly

I soldered up the PCB with the parts I got from Mouser. I realised I got one op-amp's pins flipped wrong on my design so you can see an ugly little botch to fix that! I also didn't realise the Pro-Mini's regulator can't handle >10 V so I burnt one out whilst adjusting the supply rail voltages. Lucky I had bought a pack of 3 Pro-Minis… I added a zener diode after this mistake to drop the voltage down before feeding it to the Pro-Mini. I have corrected these problems on the schematic and PCB on github and called the PCB V2.0.









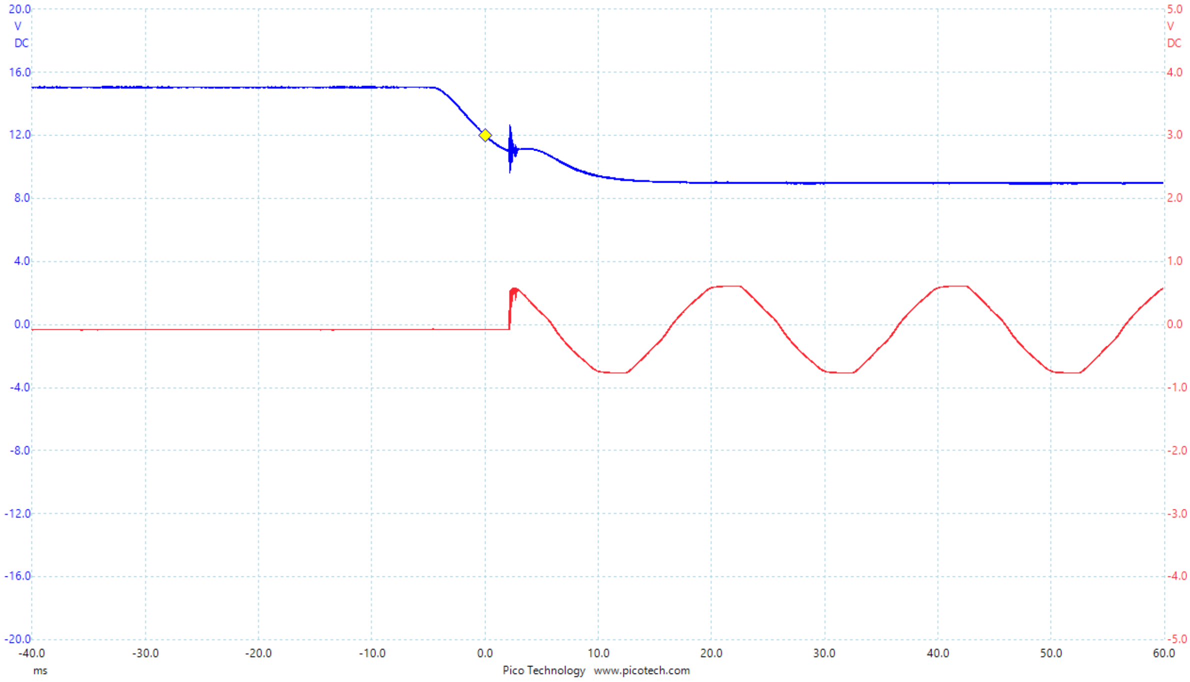
Software

I'm no software whizz. Hopefully the code is commented well enough to make sense. Find it on the GitHub repository.

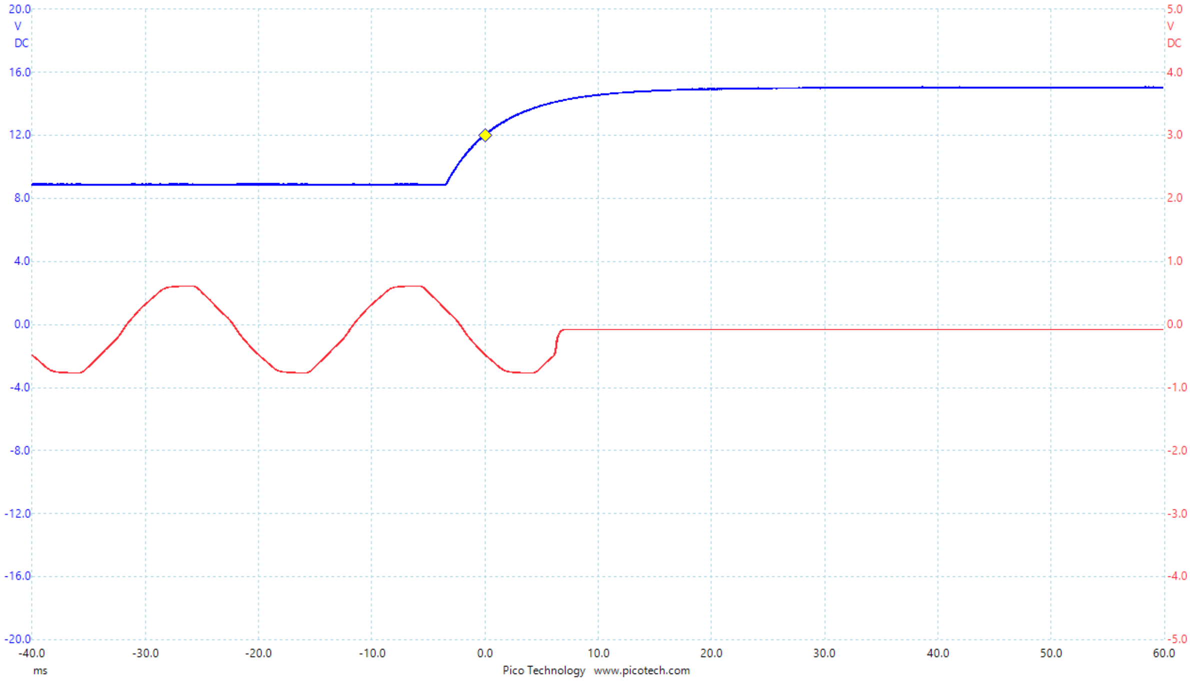
There is a self test coil on the current transformer. The software performs a self test by putting a 50Hz 5 mA square wave through the 5-Turn test coil. This essentially simulates a 25 mA ground fault. We time how long it takes for the fault to be detected. If the fault is detected within 100 ms the test is passed.

Testing

Out of interest, I measured the voltage across the relay coil when actuating. The blue trace is the coil voltage and the red trace the switched AC output. It seems to take about 5 ms to actuate the relay. The contacts obviously bounce a little bit after coming together.

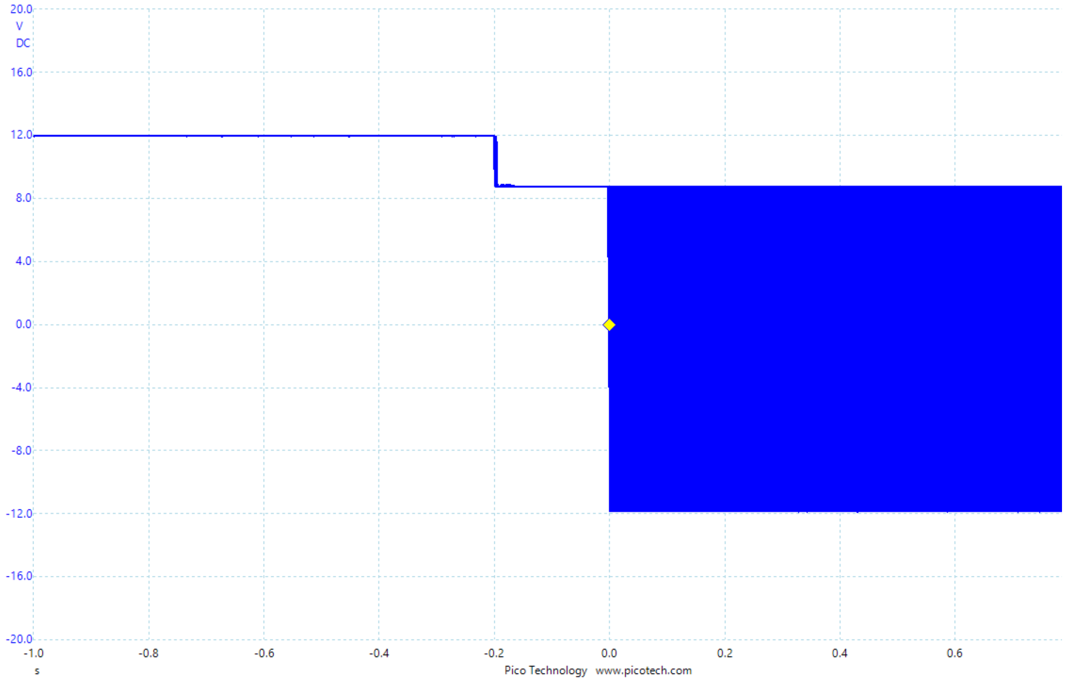


Likewise the release of the relay seems to take about 10 ms.

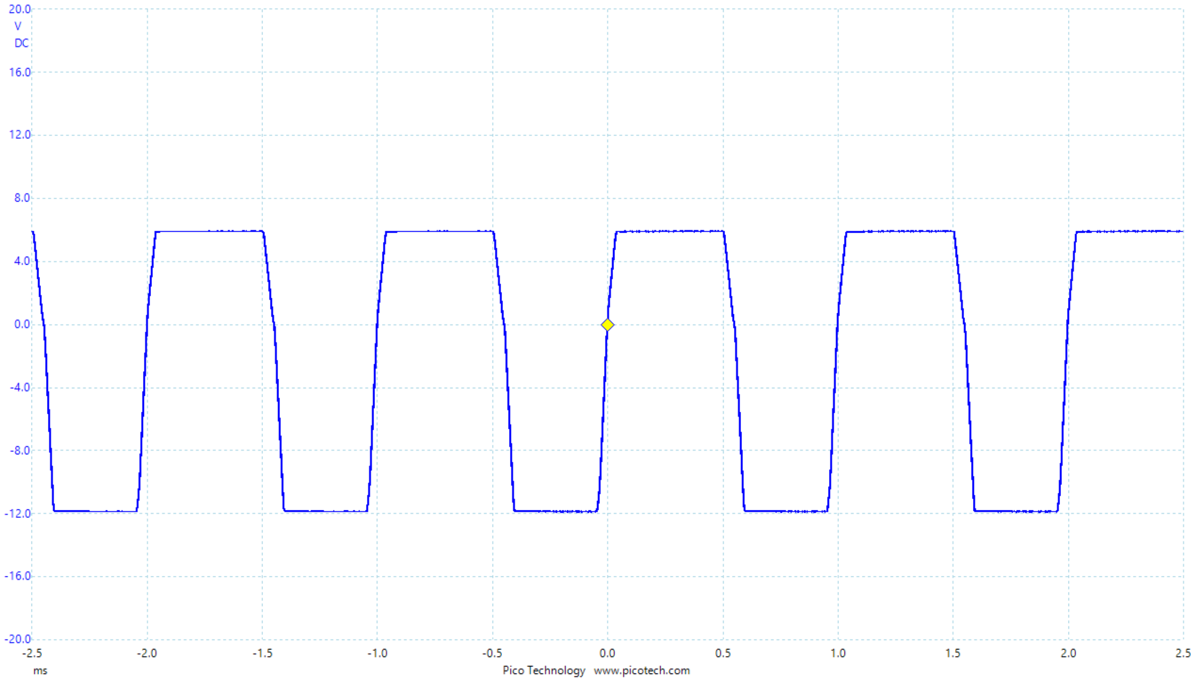


The pilot voltages are exactly as they are supposed to be. The SAE_J1772 specifications do allow +/- 0.5V from the +12V, and +/- 1V on the +9V, 6V and 3V levels so we're comfortable.

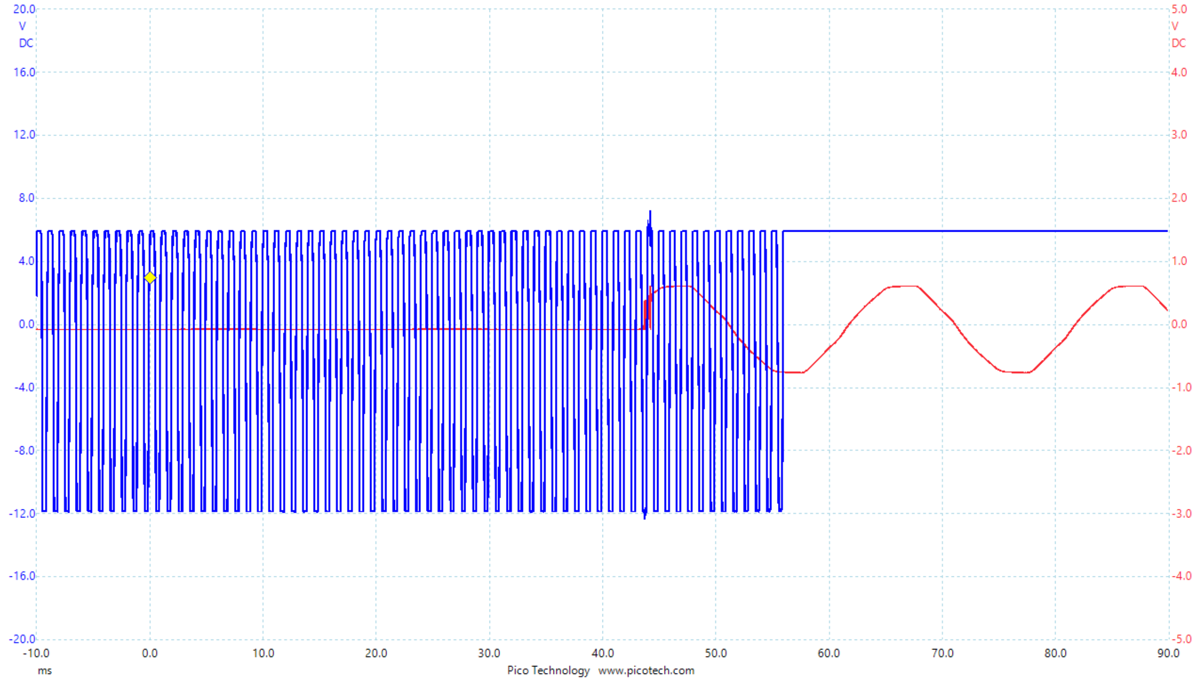
This trace shows what happens when a car is detected. The +12V DC pilot is pulled down by the car's 2.74K resistor. After a 200ms hiatus, the software switches to "State B" and starts the 1kHz PWM.



Here's the pilot when the car is charging. -12V to +6V. The software is measuring the voltages at the centres of the low and high regions.



It was important to test the ground fault detection circuit. I tested it to trip at 6 mA. Here's a trace to show the speed of a trip when a 22K resistor is connected to earth across the 240V live resulting in an 11 mA current. The detection time is 12 ms (start of the AC waveform to the rising edge of the blue waveform. Hence with the 10 ms relay release time the power will be cut in 22 ms. That's within the EV charger national specifications.



Conclusion

I'm delighted with the end result. It doesn't have an LCD screen but my car tells me how fast it's charging and allows me to configure charging times etc. I don't need any smart functionality. I have a smart meter so I know how much electricity I'm using etc too.

Total cost? Well about £200. Plus several days of hard thinking, soldering and coding. Good fun though. Let me know what you think.