My Figaro Clock Project

When we bought out Figaro back in 2016 one of the annoying little faults was the clock not working. Asking around people said, "you won't repair it and you can't buy a replacement". We were stuck with a clock that was only correct twice a day.

I do a little hobby electronics and frequently get emailed with sales pitches for new electronic components. One of these emails contained a picture of a small circular TFT display, a GC9A01. Basically, a colour led display the size of a wristwatch. After checking the specifications, I decided this was the perfect size to replace the Figaro clock.


Now I needed to find a small microcontroller to drive it. There isn't a lot of room inside a Figaro Tachometer, and I wanted to package it on a PCB on the back of the display. The Raspberry Pi Pico microcontroller had been released early in 2021 and companies were now implementing the RP2040 chip on their own development boards. The smallest I could find was the aptly named Tiny2040 made by Pimoroni.

So, I had a display and a microcontroller. Now I needed to design a circuit to connect the Tiny 2040, a power supply, a fuse, various bits of electronics to read the state of the ignition and the instrument lights, some buttons to adjust the time and to drive the display.

I didn't want the clock illuminated all the time because I wanted to minimise the power consumption, minimal though this is, and extend the life of the display. I needed the microcontroller to sense the ignition was on to then power up the display. I also wanted the display to be less bright at night, so I needed to dim the display when the car lights were switched on. All of this is done by sensing various voltages on the printed circuit board on the back of the Tachometer housing.

To prove everything would work I created a test circuit on a bread board and wrote some software in CircuitPython. The display required a bitmap of the clock face, an hour hand and a minute hand. The reason I wanted to use the RP2040 microcontroller was because it has a real time clock built in. So once powered up it keeps excellent time. The display is updated every second and moves the clock hands to denote the appropriate time.
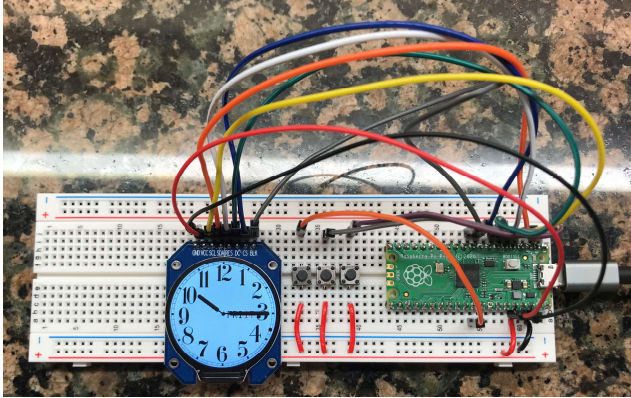

Having proved out the concept on a breadboard and a std RP PICO. I moved onto a Veroboard prototype using the TINY 2040 which I could house inside the Tacho housing.
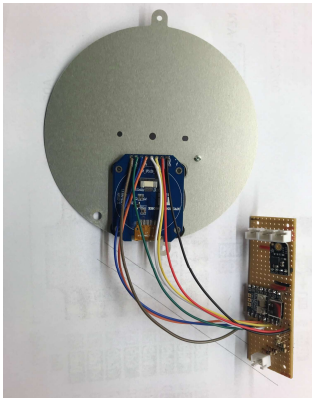
We were having work done on our Figaro at the Figaro Shop at the time and they lent me a nonfunctioning Tacho to use as a prototype. That way we could keep our car on the road while I worked on the clock. Here it is with the stripboard prototype in place.

Removing the old clock was relatively easy. Taking measurements off this I designed a mount for the clock face and the printed circuit board, that used the same fixings. This I printed in PETG. The following pictures show the design in Fusion260, the first 3D printed part and finally with the GC9A01 mounted on it.
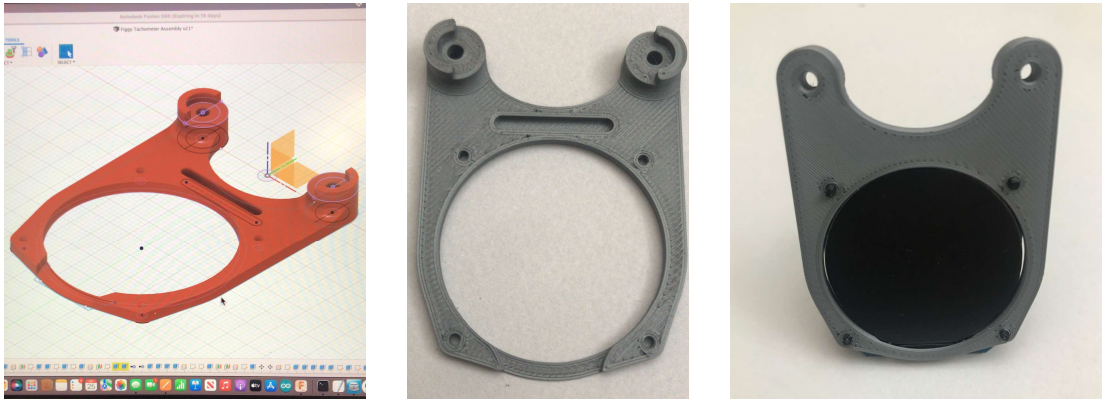
With the Veroboard prototype working I could then design the definitive printed circuit board. The World had certainly moved on from when I designed my last circuit board some 40 years ago. This time I used the amazing EasyEDA software which converts a circuit diagram into a multi layered PCB on which you can then adjust the component placement to your liking. Once satisfied you can then directly order boards online to be made on the other side of the planet. The minimum order quantity arrived a couple of weeks later.

The picture on the left is the PCB populated with components. The Tiny2040 on the left, voltage converter (12v to 3.3v) top right and two sockets, one to connect to the signal tracks on the Tacho PCB to read IGN and lights status and the other to the time adjustment buttons, forwards and backwards. The picture on the right shows the PCB mounted above the Display, itself mounted on the 3D printed carrier, eight wires connect the two electrically.
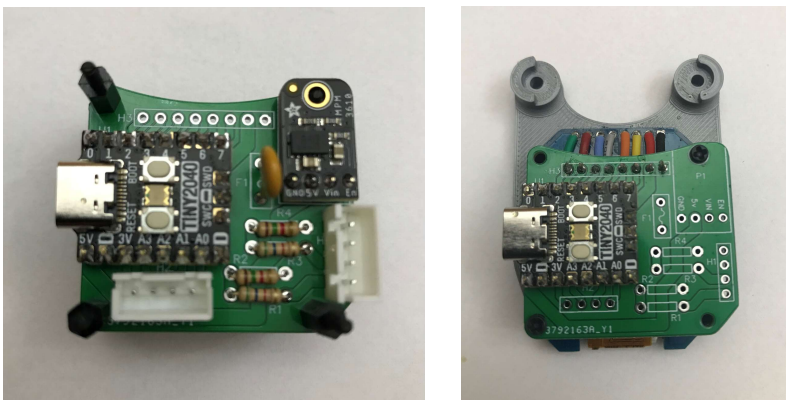
To my amazement and with huge relief everything worked when assembled and fitted in the car and it's still going strong today.

Mark Williams

2/5/2024