

A deployable control, monitoring, and logging system for the performance evaluation of TETRA handsets and meshed-network modelling

Rhys Higgins | Email: 2110763@student.uwtsd.ac.uk
BEng (Hons) Embedded Electronic Systems



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David



School of
Applied Computing

1 | Summary

1.1 | Abstract

A software controlled and vendor agnostic platform for testing TETRA PMR handsets within a real-world (outdoor) environment, utilising microcontroller (PIC18F) based nodes connected to each handset. The developed system was able to control a TETRA PMR handset, acquire the date and time from a GNSS receiver, and log data to a micro-SD card.

1.2 | What is TETRA?

TETRA PEI commands predominantly based on the AT (ATtention), otherwise known as Hayes, command set; originally this command set was developed for modem control in the 1980s using short text strings to carry out simple modem operations, such as dialing (“ATD”) and answering (“ATA”) [1].

1.3 | Project Aim

Build a field portable microcontroller platform on a PCB able to establish, maintain, and test a TETRA PMR network utilizing the extended TETRA AT command set, able to log to a platform agnostic FAT32 filesystem [2].

1.4 | Project Objectives

- Design a PCB with RS232/RS485 interfaces needed for TETRA PEI .
- Integrate AT commands into the C-language codebase.
- Log network data to FAT32 SD card via SPI.
- Implement serial interfaces (I2C, UART, SPI) using interrupt-based code.

1.5 | Hardware/Software Tooling

- MPLAB X IDE, PICKit 4 ICSP, CCS C Compiler, FAT.c library.
- KiCad V7.0 PCB designer, Proteus Professional 8
- Hot Air Soldering Station and Fine Tip Soldering Iron.
- TETRA PMR handset.

2 | Implementation

2.1 | Hardware Development

The hardware design was conducted in KiCad, with a block diagram used to inform the design of a hardware schematic, which was then translated into a PCB as industry standard Gerber files for manufacture [3]. The PCBs were manufactured by a supplier and then populated with components by the author.

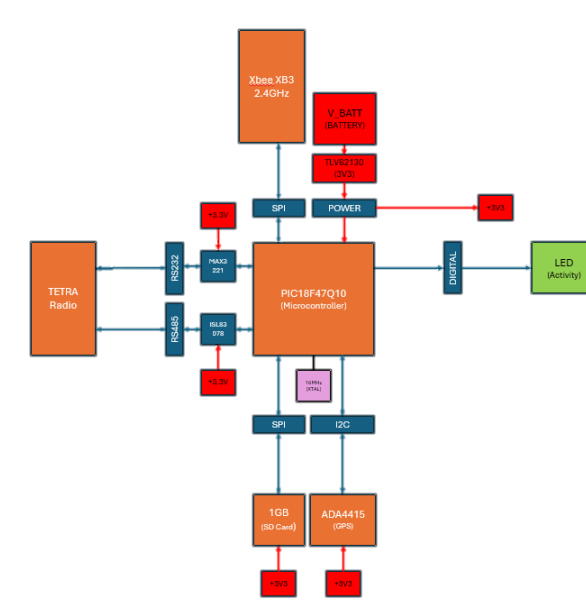


Figure 1 – Block Diagram

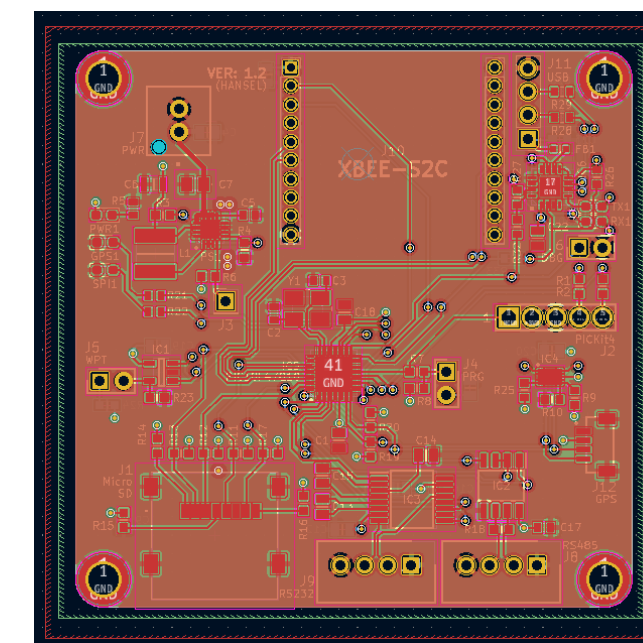


Figure 2 – PCB Design

2.2 | Software Development

Functions were developed in C, using MPLAB X, with the system functionality being controlled by Interrupt Routines to carry out system functions determined by the data sent and received via USB UART and ZigBee. The ETSI TS 100 392-5 [4] document was used to develop the AT command library for the TETRA radios.

Due to the use of software UART ports, the “INT_RDA” interrupt that would typically be used to capture Incoming UART packets was not able to be used [5], instead the “INT_TIMER0” interrupt was used to poll the ports at 68us intervals.

```
void main_12c_bus(void)
{
    int status;
    int counter = 0;

    printf(DEBUG, "\n\rStarting 12c bus\n\r\n\r");

    // Test all devices addr from 0x0 to 0xFF.
    // Check for response from device.
    for(int i = 0x00; i < 0xFF; i++)
    {
        status = get_i2c_status(i);
        if(status == TRUE)
        {
            //Not this for reading external command
            //Send UART/TX interrupt for data command
            //Send commands to device/Device function
        }
    }

    if(status == TRUE)
    {
        UART2_send() = Sync(DEBUG);
        if(UART2_recv() == 0x)
        {
            UART2_send(12C_CMD);
            printf(DEBUG, "\n\r\n\r", UART2_recv());
        }
        if(status == TRUE)
        {
            UART2_send() = Sync(DEBUG);
            if(UART2_recv() == 0x)
            {
                UART2_send(12C_CMD);
            }
        }
    }
}
```

[illegible]

Figure 3 – Code Snippets

3 | Results

3.1 | Testing & Conclusion

The project was mostly successful; however, the full TETRA AT command set was not fully implemented due to time limitations – primarily due to overruns during the hardware manufacture.

The PCB prototype system was able to read GPS data as NMEA strings, see Figure 5, and store create files and write them to the FAT32 filesystem on disk, see Figure 7, among communicating with a Hytera TETRA PMR handset.

The code functioned well, and the time-based interrupt functions also were able to capture packets received over the software UART2/3 interfaces.

The design itself was successful and with further development to the codebase, the existing hardware (without revisions, see below) would be able to provide greater functionality and therefore value.

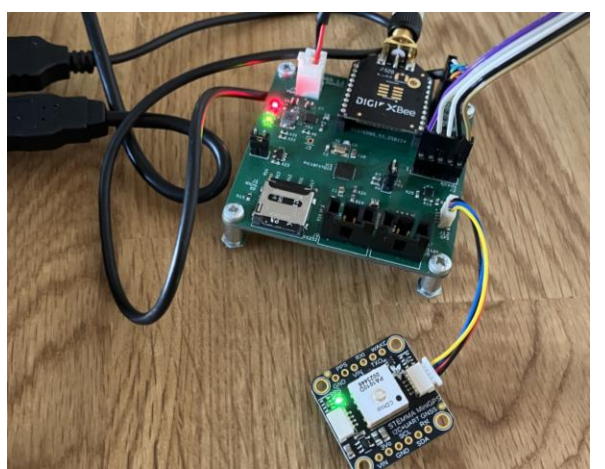


Figure 4 – System Working

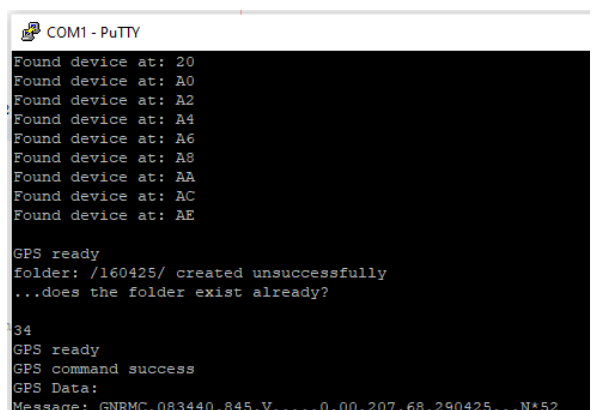


Figure 5 – System Output

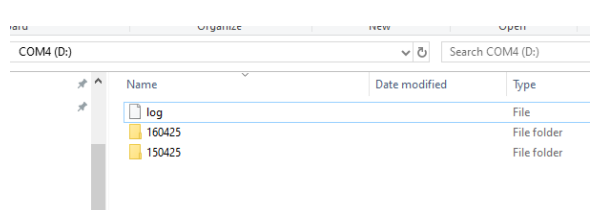


Figure 7 – Files on disk

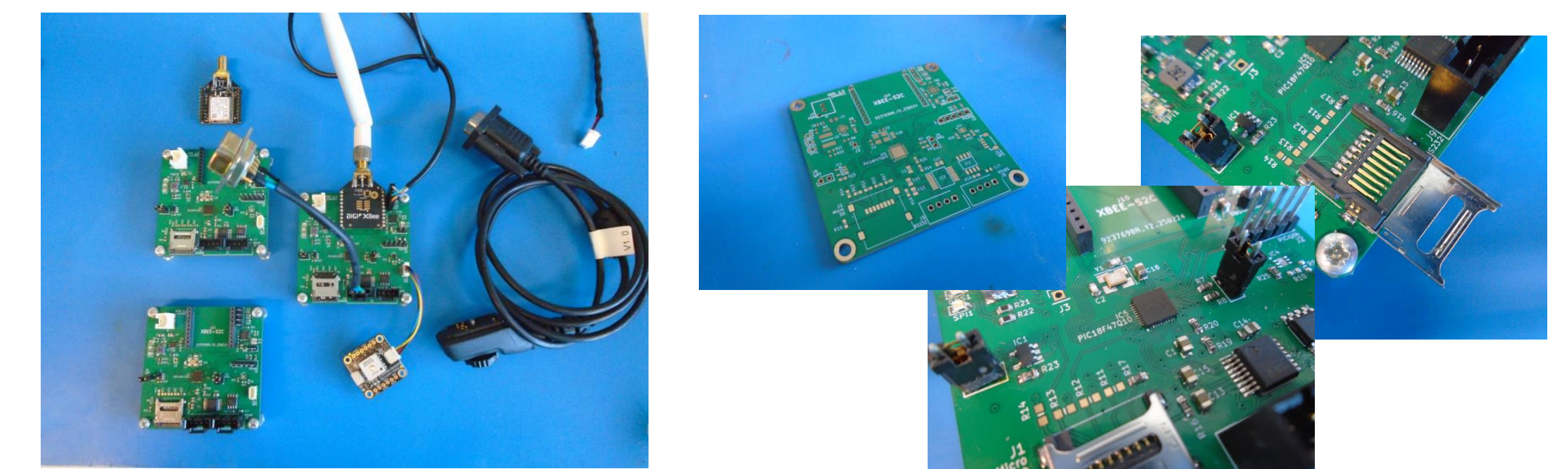


Figure 6 – Final Hardware

3.2 | Future Development

The system was overall successful, however there are several lessons which were identified.

- Add recycling labelling to PCB silkscreen for WEEE compliance [6].
- Add a POWER/RESET button to enable easier testing.
- Utilize an RS-232/RS-485 multiprotocol transceiver rather than separate RS-232/RS-485 transceivers.
- Add voltage monitoring to +VIN of +3V3 buck converter.
- Fix the PICKit4 ICSP header as the PGC and PGD are flipped.
- Enable AES encryption on ZigBee modules.
- Utilize PIC power-saving modes, such as DOZE [7].
- Develop the command handler function on the modules to use string-based messages.

References

- [1] S. Sunny, "AT Commands Tutorial," Cavli Wireless, 2024. [Online]. Available: <https://www.cavliwireless.com/blog/nerdiest-of-things/an-introduction-to-cellular-at-commands.html>
- [2] OSDEV.org, "FAT," 2024. [Online]. Available: <https://wiki.osdev.org/FAT>
- [3] Z. Peterson, "What Is A Gerber File and How Are They Used in the PCB Fabrication Process?," 2024. [Online]. Available: <https://resources.altium.com/p/what-gerber-file-pcb-fabrication-process>
- [4] ETSI, "ETSI TS 100 392-5 V2.4.1 (2016-03), Technical Specification," 2016. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/100300_100399/10039205/02_04_01_60/ts_10039205v202401p.pdf
- [5] Unknown, "#INT RDA," CCS Forum, 2019. [Online]. Available: <https://www.ccsinfo.com/forum/viewtopic.php?t=57986>
- [6] Europa.eu, "WEEE label," 2025. [Online]. Available: https://europea.eu/your/europe/business/product-requirements/labels-markings/weee-label/index_en.htm
- [7] M. I. R. Salimabao, "TB3144: Doze, Idle and PMD Features of 8-Bit PIC® Microcontrollers," 2015. [Online]. Available: <https://www.1microchip.com/downloads/en/appnotes/90003144ab.pdf>