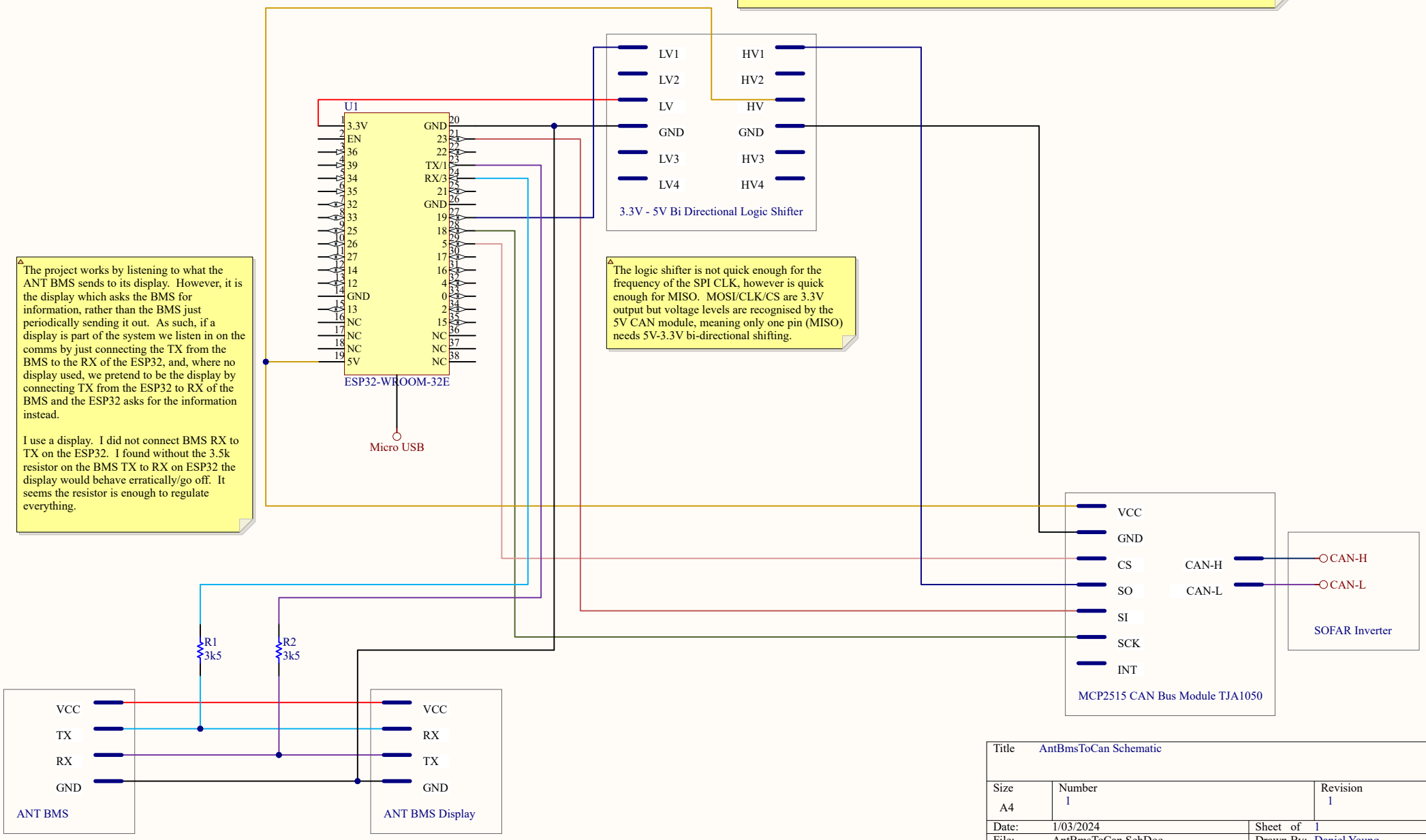
RS232 Hardware Serial Pins:
ESP32 PIN 1 - Serial Transmit (Output from ESP32)
ESP32 PIN 3 - Serial Receive (Input to ESP32)

SPI (Serial Peripheral Interface) Pins:
ESP32 PIN 23 - MOSI - Master Out, Slave In - Can be 3.3V as it is an output.
ESP32 PIN 19 - MISO - Master In, Slave Out, Needs 5V-3.3V protection as it is an input.
ESP32 PIN 18 - CLK - Clock signal from ESP32, Can be 3.3V as it is an output.
ESP32 PIN 5 - CS - Chip Select - Governs communication (wakes up the slave), Can be 3.3V as it is an output.



The project works by listening to what the ANT BMS sends to its display. However, it is the display which asks the BMS for information, rather than the BMS just periodically sending it out. As such, if a display is part of the system we listen in on the comms by just connecting the TX from the BMS to the RX of the ESP32, and, where no display used, we pretend to be the display by connecting TX from the ESP32 to RX of the BMS and the ESP32 asks for the information instead.

I use a display. I did not connect BMS RX to TX on the ESP32. I found without the 3.5k resistor on the BMS TX to RX on ESP32 the display would behave erratically/go off. It seems the resistor is enough to regulate everything.

The logic shifter is not quick enough for the frequency of the SPI CLK, however is quick enough for MISO. MOSI/CLK/CS are 3.3V output but voltage levels are recognised by the 5V CAN module, meaning only one pin (MISO) needs 5V-3.3V bi-directional shifting.

Title	AntBmsToCan Schematic	
Size A4	Number 1	Revision 1
Date: 1/03/2024		Sheet of 1
File: AntBmsToCan.SchDoc		Drawn By: Daniel Young