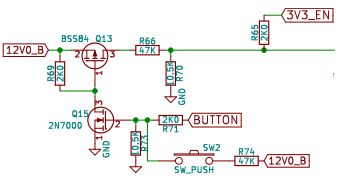
V2X v1.2 Hardware Description

Author Jesse Banks

The design of the V2X board has an Atmel microcontroller at its heart. When the v2x board first receives power it should remain in an off state. the board has a singular button with dual purposes, starting power on the board and acting as a soft button for the sequencing chip (and vicariously the host computer software).



pressing the button will temporarily enable the 3V3 power supply, providing power to the atmel chip. The first action of the Atmel needs to be resetting shift registers to the default 0 state, then sending the 3V3\_EN signal to keep the 3V3 Volt Power System alive. Once the button is released, the 3.3V power supply will shut off if the 3V3\_EN is not pulled high by the Shift registers. There are more power modes described later.

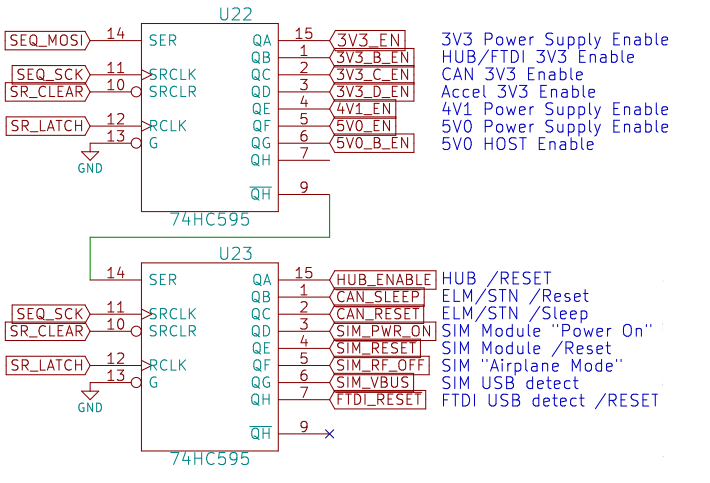
The BUTTON signal is routed to the ATmega Port A0. It produces a positive logic signal (pin goes high on press).

SEQ\_CHGPUMP operates a diode switched capacitor voltage doubler. this pin should toggle at >10Khz, and 100kHZ is better. Without running the charge pump the LEDs may not light.

SEQ\_LED1, 2, 3 are all negative logic signals, a logic low causes the light to turn on. They can be pulsed at 10’s of Khz to create fading.

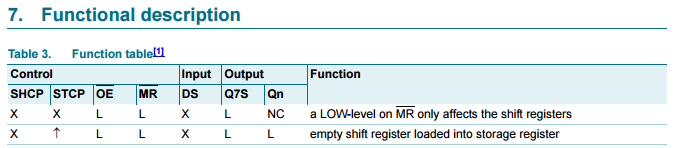
Shift registers

The method for controlling power switches and reset signals throughout the board is handled by a pair of 74HC595 serial shift registers. the shift registers use the ATmega SPI peripheral bus signals SCK and MOSI, along with dedicated clear and latch signals. Output Enable is signal “G” it is tied to ground so that the outputs are always on, and never high impedance.



[The shift register data sheet is here](http://www.mouser.com/ds/2/302/74HC_HCT595_Q100-269352.pdf)

The running state of SR\_CLEAR is ‘1’ (not clear). The shift register reset procedure involves setting SR\_CLEAR to ‘0’ (this clears all the shift registers) then cycling SR\_LATCH 0->1->0 (moves the clear SR value to the output buffers) and setting SR\_CLEAR to ‘1’ again. this operation loads all ‘0’s to the outputs and puts the control signals back in their default states.





Power

Power switches allow for independently powering the ATmega, accelerometer, CAN bus controller, GSM/GPS modem, USB hub and FTDI along with the HOST computer itself. All of the power signals are grouped into the shift register closest to the ATmega, though this implies it receives the data sent last in the SR packet.

Power modes:

3V3\_EN = ATmega

Plus 3V3\_B\_EN = HUB and FTDI

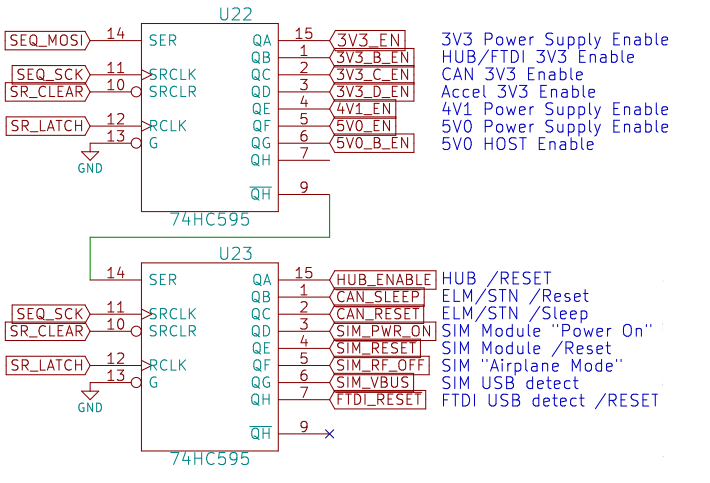
Plus 3V3\_C\_EN = CAN (Also needs 5V0\_EN)

Plus 3V3\_D\_EN = Accelerometer

4V1\_EN = GSM and GPS and ~2.3V SMS sleep for ATmega

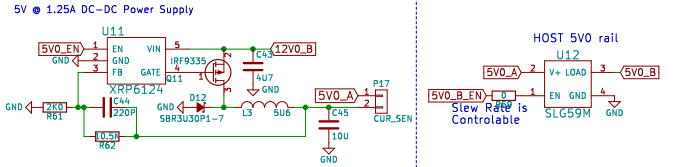
5V0\_EN = CAN

Plus 5V0\_B\_EN = HOST

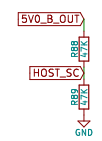
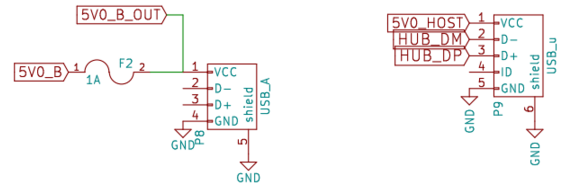


HOST Power

A ‘wake” event may triggered by the real time clock alarm, button press, or GSM/SMS received. The 5V0\_EN signal is set high to start the 5volt power supply, U11. After a short delay (100ms guess), the 5V0\_B\_EN signal is set high to provide the host with power (5V0\_B) through power switch U12.



Normal usage has the power for a Rpi2B HOST being delivered by USB cable from P8 mounted to the bottom side of the V2X PCB. The HOST could also be beaglebone, minnowmax, Rpi 1, etc.

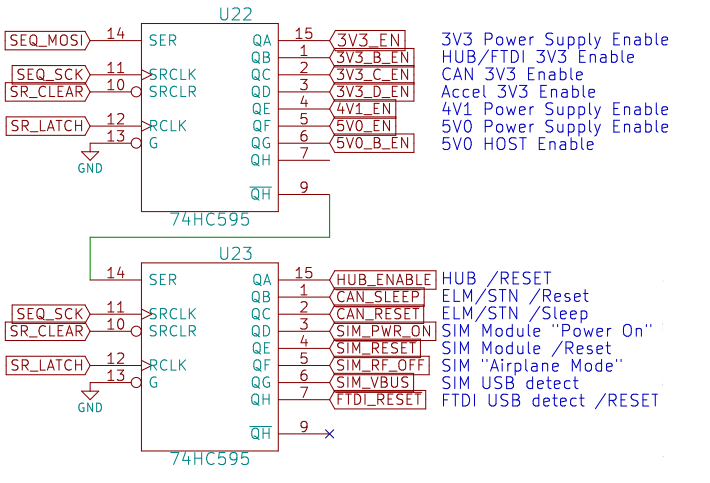


While 5V0\_B is powered, HOST\_SC can be monitored to determine if the is causing a short circuit, setting the poly fuse open. if HOST\_SC is read as a logic ‘1’ the host is receiving power. A logic ‘0’ indicates the fuse is protecting the power supply. P9 is USB back from the HOST.

Once the HOST is started the ATmega receives AT commands (serail from FTDI or USB from HUB) that allows the HOST to modify the power modes (maybe just envoke canned sequences).

Enable and Reset signals

Most of the reset and sleep signals are grouped into the shift register farthest from the ATmega, this implies it receives the data sent first in the SR packet.



HUB ENABLE - Likely not used, routes to R18/R19 switch. Default config (R19=JMP) has the 5V0\_HOST (USB Vbus) signal perform this action automatically.

CAN\_SLEEP

External sleep control input. When enabled in firmware, puts the device into low - power sleep mode. Polarity of this pin can be configured in firmware; default configuration is active low. Internal pull-up to VDD is enabled by default, but can be disabled in firmware. When STN11xx senses a logic low on the SLEEP pin, it immediately aborts any OBD reception in progress, or monitoring command that is active at the time, and prints the command prompt. It then monitors the SLEEP input and enters the Power Save mode if the minimum low time (specified by the STSLXST command) is satisfied.

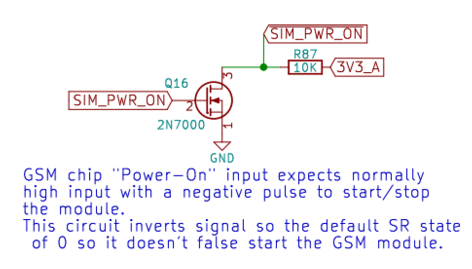
CAN\_RESET

Device reset input. A logic low pulse (min 2μs) on this pin will reset the device. Apply a continuous logic low to hold the device in reset.

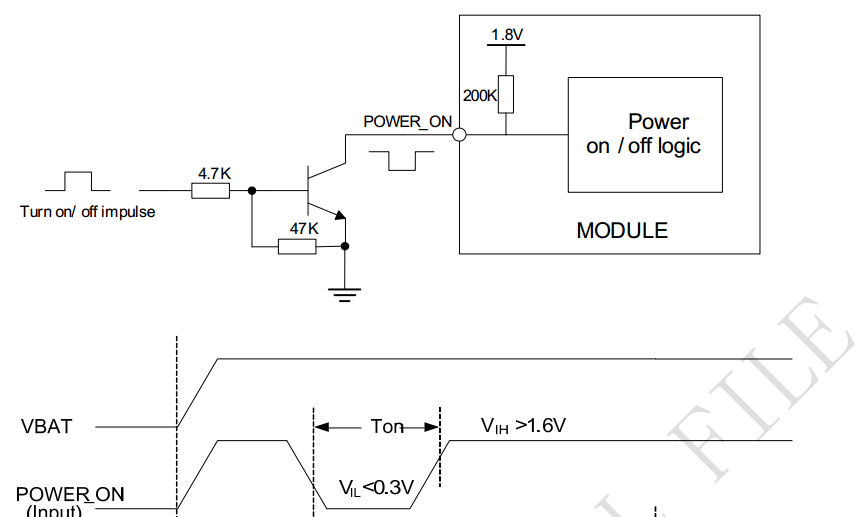
SIM\_PWR\_ON

The SIM5320 module expects a negative pulse to start the module. Q16 inverts a positive pulse to create this signal. This way a positive pulse from the shift register creates the signal the SIM is expecting. Also, the Shift register default 0 state should not trigger the SIM module to start falsely.

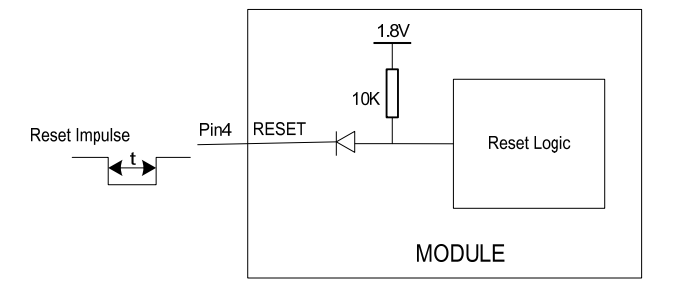
This circuit makes SIM\_PWR\_ON become SIM\_PWR\_ON



Here is a quick clip of the SIM module data sheet:



SIM\_RESET



Should be high to operate the SIM5320 module. Only necessary to use as reset command if SIM stops responding to AT commands.

SIM\_RF\_OFF (GPIO4)

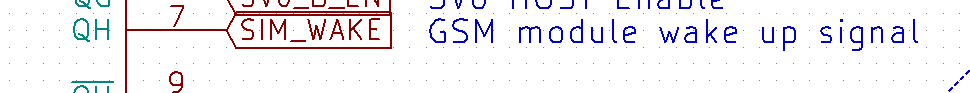
Positive logic RF operation control. A low signal causes the module to stop transmitting to observe “being on an airplane.” Set signal High for normal opperation

SIM\_VBUS

Likely not used, routes to R10/R11 switch. The signal routing controls SIM\_USB\_DET which the SIM5320 uses to detect USB attachment. Default config (R11=JMP) has the HUB\_PWR\_PRT2 (Hub port active status) signal will perform this action automatically.

/FTDI\_RESET

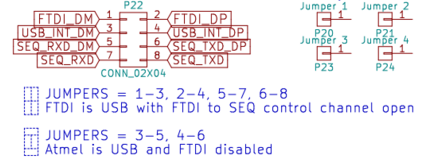
Likely not used, routes to R2/R3 switch. The signal routing controls /RESET pin on FT4232 which is used to detect USB attachment. Default config (R3=JMP) has the HUB\_PWR\_PRT2 (Hub port 2 status) signal will perform this action automatically.



SIM\_WAKE is the only enable signal driven by the shift register closest to the ATmega. The signal wakes the GSm module from sleep.

USB Data

HOST Control of the V2X is handled completely over USB. As it enters the board it hits a 2 port hub. the hub sends the signals to the SIM5320 module and the FDTI4232 or ATmega through jumper settings on P22 (2x4 pin header)



A goal of the ATmega firmware is to replace the FTDI with it’s own multipoint USB device inside the ATmega. both configurations should operate effectively the same as used by the HOST.

Both configurations make use of the 5 endpoint interface provided by the SIM5320 module. they provide GPS, modem and SMS information, with a simple relay through the USB hub.

The other 3 roles are Accelerometer, CAN and power sequencing control. we would use the FTDI as a plug-n-play 3-port serial pass through device until the atmel drivers are loaded.

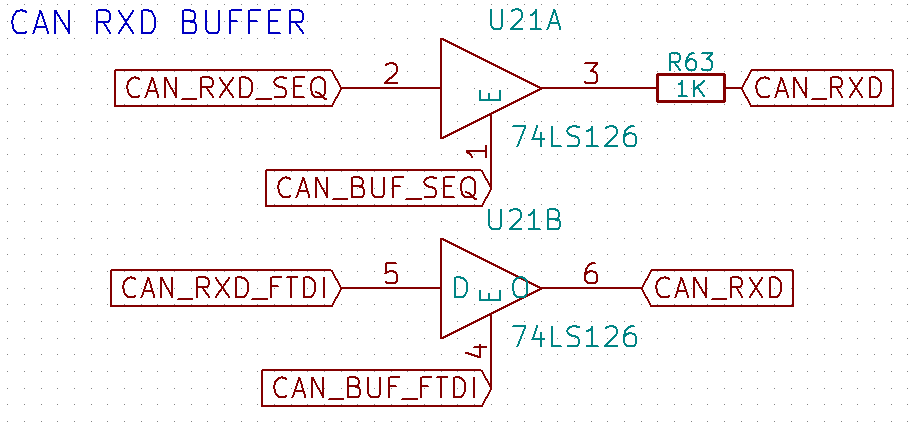
With the FTDI in use, the CAN communications use one of the serial endpoints in the FT4232. the ATmega uses one channel to pass motion data and receive accelerometer config data (sample rate). A second serial endpoint is used to carry the power sequencer AT command interface.

If the FTDI is bypassed the ATmega pins that proved the serial command path to the FTDI become the USB data pins. The firmware will have to create a passthrough to the CAN device just as the FTDI did. And have two other endpoints for the ACL and SEQ command interfaces.



Serial Data

The ATmega is serially connected to the SIM5320 module, FTDI, and the CAN controller (STN1110). This system has multiple options for configuration. The variation is to support removing the FTDI from the system.



If the FTDI is used in the system, the ATmega has to control the signal routing for the CAN device. CAN\_RXD\_SEQ and CAN\_RXD\_FTDI are mutually exclusive signals. they each control the output enable signal to the CAN device receive pin. the signals should not both be high as the outputs of two buffers will fight. R63 is in place to keep them from hurting each other. the signal from the FTDI will win if both buffer outputs are active at the same time.

In active operation the HOST will be speaking to the CAN device to snoop the car. it will also be speaking to the ATmega through parallel channels for; gather accelerometer data and issue commands to the “power sequencer” function in the ATmega. this architecture was chosen to create easy differ entiation through unique USB endpoints.

If the FTDI is bypassed the ATmega will be performing the functions of the FTDI. It will be creating 3 “serial” endpoints for the HOST in with the same appearance to the host. The CAN serial buffer can be hard set to enable only the ATmega and disable the FTDI TXD.

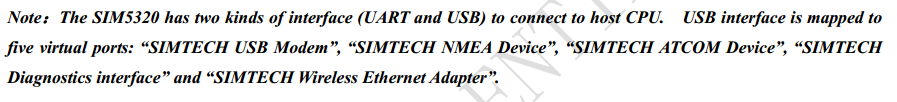
The ATmega will change rolls when the host is on and off, acting as a serial pass through to the HOST, or it will activate the CAN controller periodically to snoop for ignition detection. The accelerometer and Power sequencer USB endpoint stay completely inside the ATmega.

SIM5320

The SIM5320 gets its subscription information from a sim card just like any other cell phone. The sim card slot has a switch for detecting when the card is inserted. The SIMCARD signal, Port D5, uses negative logic. When the pin is high there is no sim card in the socket. The pin should be configured with a pullup resistor, as an inserted card will close a switch to ground. For highest power efficiency the port-pullup should only be enabled several microseconds before a reading is taken, and disabled immediately afterward.

The GPS System is backed up by a super capacitor for rapid satellite acquisition. First acquisition can take 30+ seconds or more, subsequent acquisitions should be <1 second. The super capacitor should keep the clock running for 5-10 hours based on limited info.

The primary means of control is by USB connection. Through the USB are five virtual comports.



Simcard operation should be handled automatically. Card Logic voltage detection is also automatic.

the line of **misinformation** (don’t trust below this line)

<http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf>