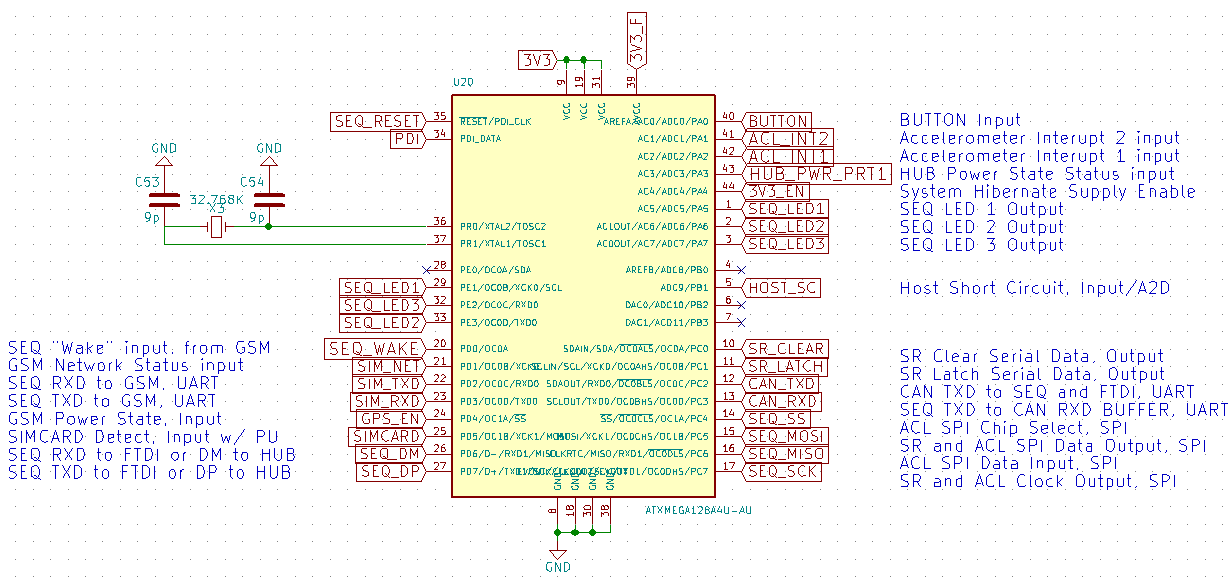
V2X v2 Hardware Description

Author Jesse Banks

## Architecture

The design of the V2X board has an Atmel microcontroller at its heart. The atmel has 4 primary jobs:

1. control system/sub-system power states and sequencing,
2. creates a control/reporting interface (AT style) for each system,
3. serial adapter/passthrough for the ELM327/STN1110 chip,
4. samples and generates a serial stream from accelerometer data.



## Button

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 and soft button. pressing the button will temporarily enable the 4v1 power supply, passing power through U12 and 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 4V1\_EN or 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 4V1\_EN or 3V3\_EN is not pulled high by the Shift registers.

After starting power, the button it will transition to a soft button function. The BUTTON signal is routed to the ATmega Port A0. It produces a positive logic signal (pin goes high on press). The command interface will report press and release events through the CMD usb port, and will have the following forms:

Press event: Sent to CMD interface: ”BTN:>PRESS\n”

Release event: Sent to CMD interface: ”BTN:>RELEASE:(seconds pressed)\n”

Here is a terminal report:

BTN>:PRESS

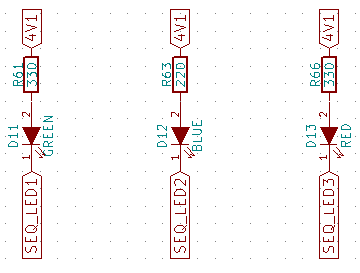
BTN>:RELEASE:1

BTN>:PRESS

BTN>:RELEASE:4

## LED

SEQ\_LED1, 2, 3 are all negative logic signals, a logic low causes the light to turn on. They can be pulsed at 1-10’s of Khz to create fading using OC0B, OC0C and OC0D (PE1-3).

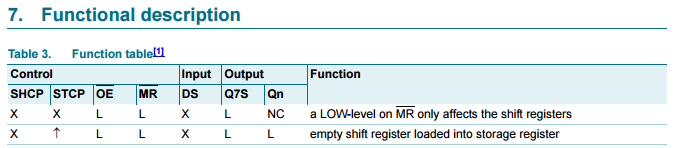


## Shift registers

The method for controlling power switches and reset signals throughout the board is handled by a 74HC595 serial shift register. the shift register uses 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 operational 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.





3V3\_EN: (not from SR) Low power supply for idle and sleep modes.

4V1\_EN: main power supply, needed for GSM and GPS.

5V0\_EN Starts switching supply for 5V0 power rail. CAN and HOST

5V0\_B\_EN: load switch powering HOST

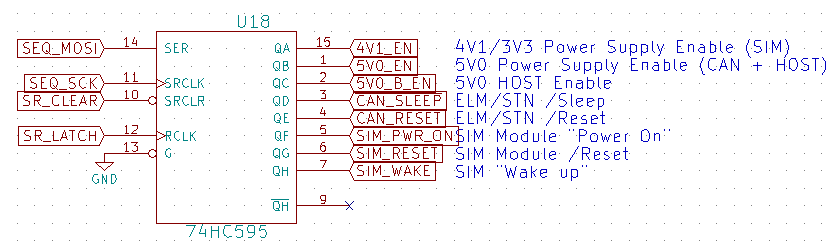
/CAN\_SLEEP: suspend signal to CAN device (active low)

/CAN\_RESET: reset signal for CAN device (active low)

SIM\_PWR\_ON: wake-up/shut-down pulse signal

/SIM\_RESET: causes the SIM GSM/GPS module to reset

SIM\_WAKE: cycles SIM module between sleep and wake states (not implemented yet)

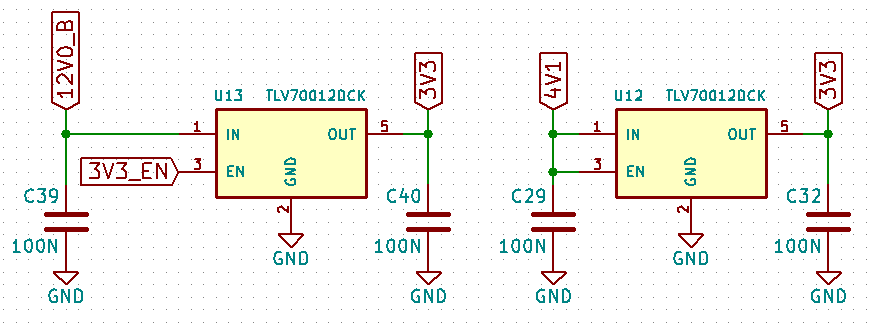


### 4V1\_EN, Main Power

Power switches allow for controlling power to the ATmega, accelerometer, CAN bus controller, GSM/GPS modem, USB hub and FTDI along with the HOST computer itself.

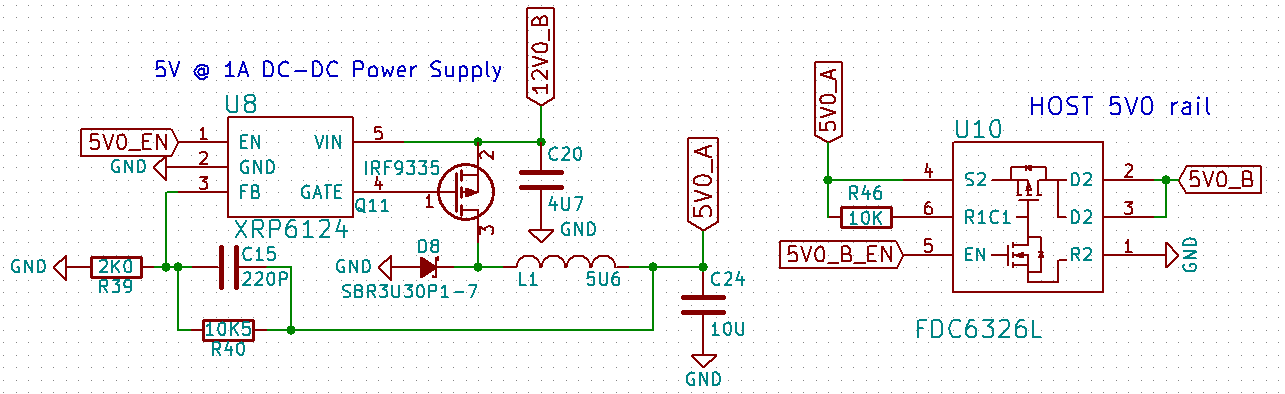
The 3V3 power rail is critical to the ATmega for operation. The 3V3 power can come from 2 sources, but should only be using one at a time (minimize overlap). Identical LDO linear regulators sourcing from 4v1 and filtered 12V are available.

Since linear regulation is wasteful, 4V1 should be used whenever possible. For this reason the enable pin of U12 is tied to 4V1 and will automatically power the 3V3 rail if the 4V1 rail is powered. In this case U13 should be disable by holding 3V3\_EN low. If 4V1 is being powered off for sleep mode then 3V3\_EN (port A0) should be set high immediately prior to keep the ATmega alive.

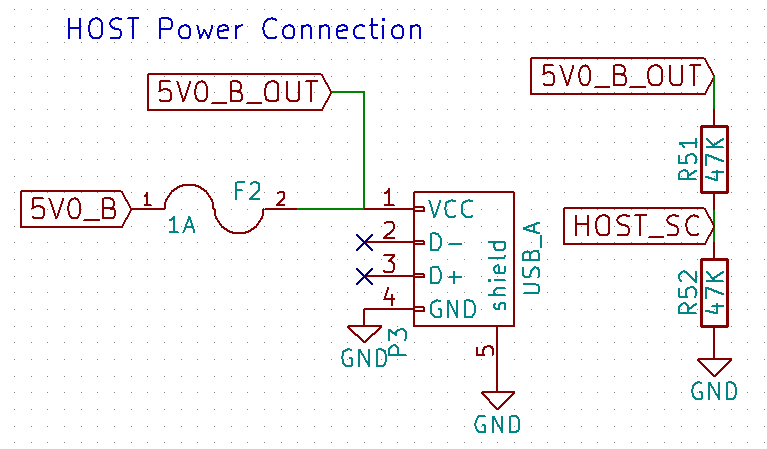


### 5V0\_EN and 5V0\_B\_EN, CAN and HOST Power

The 5V0\_EN signal is set high to start the 5volt power supply, U8. After a short delay (100ms guess) for supply stabilization, the 5V0\_B\_EN signal is set high to provide the host with power (5V0\_B) through power switch U10.



The HOST would typically be a Raspberry Pi 3, but it could also be a beaglebone, minnowmax, Rpi 1/2, etc. The host receives power from V2X port P3, the USB-A Female port.



While 5V0\_B\_EN is active, HOST\_SC could be monitored to determine if there is a short circuit or overload, 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. Alternately ADC9 can take an analog sample possibly indicating power usage without overload, nominal values are unknown.

Once the HOST is started the ATmega receives AT-like commands that allows the HOST to modify the power modes and access other features.

### CAN\_SLEEP

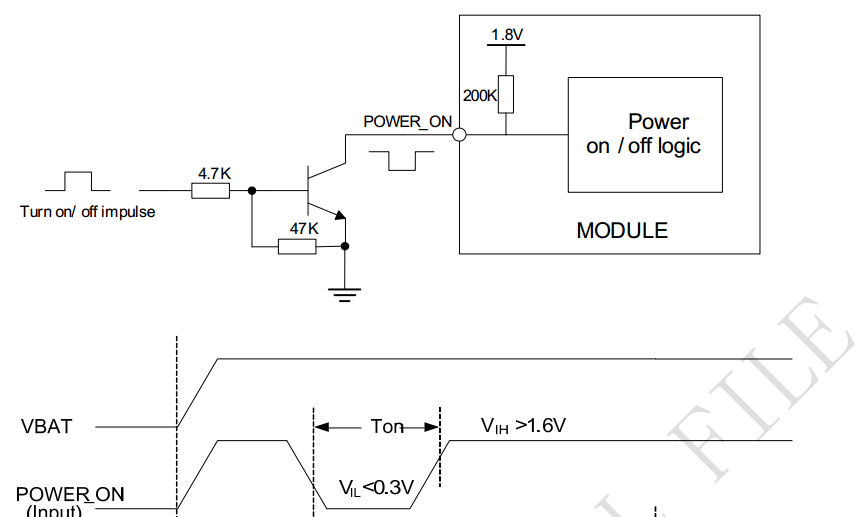
External sleep control input. When enabled in firmware, puts the device into low - power sleep mode. Polarity of this pin can be configured in firmware; default configuration is active low. Internal pull-up to VDD is enabled by default, but can be disabled in firmware. When STN11xx senses a logic low 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

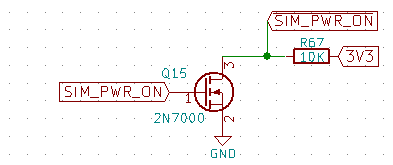
Device reset input. A logic low pulse (min 2μs) on this pin will reset the device. Apply a continuous logic low to hold the device in reset.

### SIM\_PWR\_ON

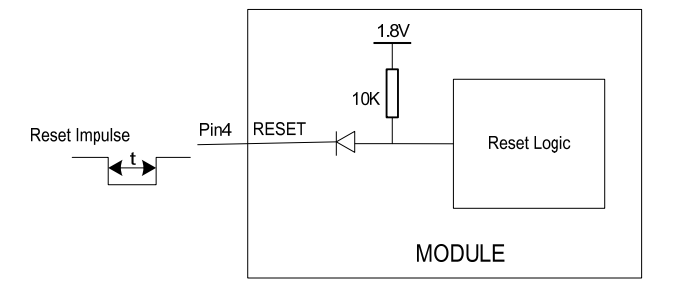
The SIM5320 module expects a negative pulse to start the module. Here is a quick clip of the SIM module data sheet:



Q15 transfers a positive pulse from the shift register into the negative signal the SIM is expecting. Also, the Shift register default 0 state should not trigger the SIM module to start falsely. This circuit changes SIM\_PWR\_ON into SIM\_PWR\_ON



### 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\_WAKE

The signal wakes the GSM/GPS module from sleep. Not currently implemented, likely requires configuration of SIM module to use.

## Data Paths

HOST Control of the V2X is handled completely over USB. As USB enters the board it lands on a 2 port hub. the hub sends the USB signals to the SIM5320 module and the ATmega.

The SIM produces a 5x USB endpoint interface. they provide the NMEA GPS stream, virtual modem, SMS and GSM control.

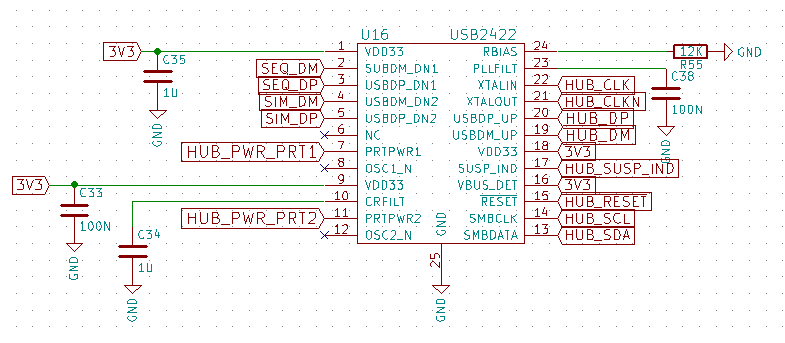
The ATmega firmware performs 3 roles (with 3 corresponding USB endpoints): Accelerometer stream, CAN interface serial bypass and system level power sequencing control.



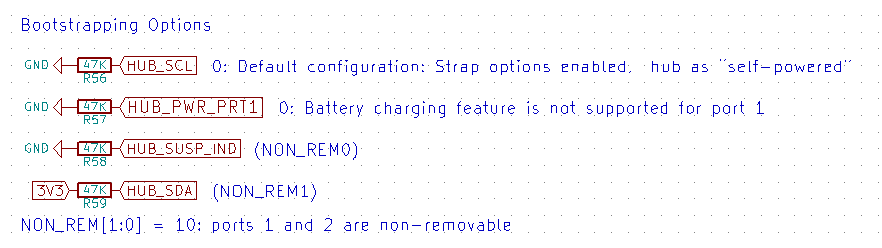
## USB Hub

The design uses a USB2422, 2-port automatic hub. [Datasheet](http://www.mouser.com/ds/2/268/2422-319423.pdf)

HUB\_RESET is automatically pulled low by the chip. It is released by an upstream USB Vdd supply (HOST) connection.



With the default board configuration the hub should activate and operate automatically as a bus powered 2-port hub. Logic states on several pins are sampled when the 2422 leaves reset (signal goes high) to determine the mode of operation.



Non default configurations allow for complete setup and configuration of the hub through SMBUS (I2C) communications. See the datasheet for all the confusion you can handle.

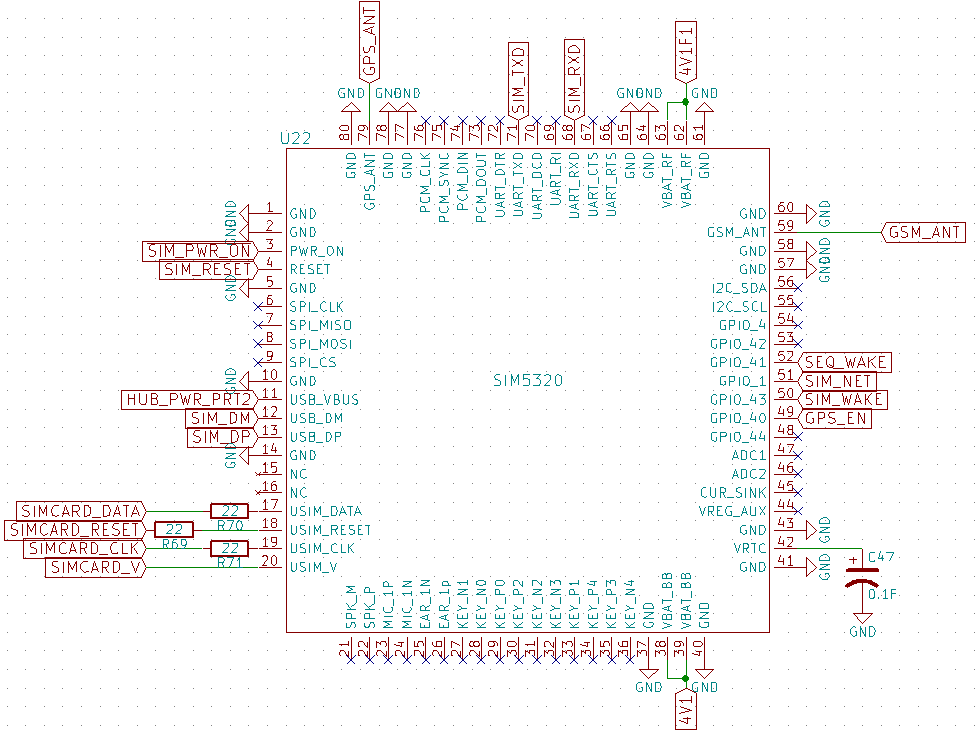
## Serial Data

The ATmega is serially connected to the SIM5320 module and the STN1110. This system has multiple options for configuration.

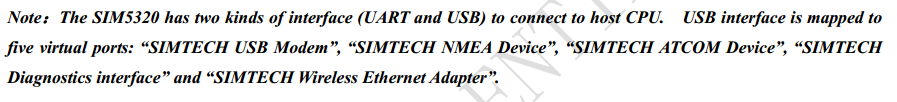
The driver in the ATmega understands when the USB passthrough channel is open to the HOST. While the HOST is offline (and special modes) the Atmel can directly access the CAN device to search for ignition evidence.

## SIM5320

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 GPS clock running for 36+ hours based on limited info.

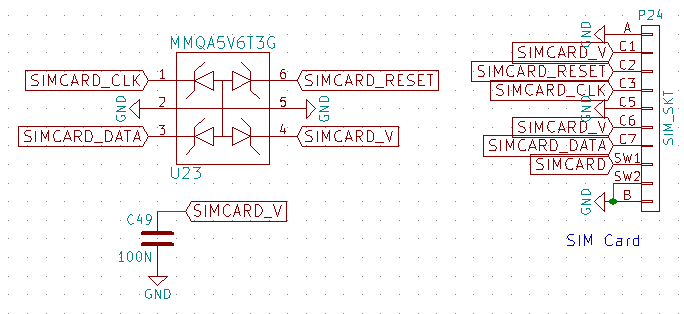


The primary means of controlling the SIM5320 is by USB connection (SIM\_DM, SIM\_DP). All functions are mirrored onto a single serial channel with functional multiplexing. The serial method is far more complicated, but gives access to the ATmega while the host is off. Excerpt from the data sheet.



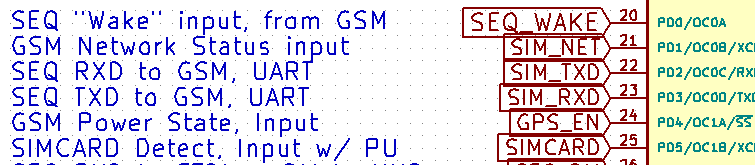
USB mounting is automatically triggered by HUB\_PWR\_PRT2 signal, generated when the hub successfully mounts to the HOST. USB\_VBUS is low otherwise and the USB port is deactivated.

The SIM5320 gets GSM subscription information from a sim card just like any other cell phone. Sim card Card Logic voltage detection and operation is handled automatically.

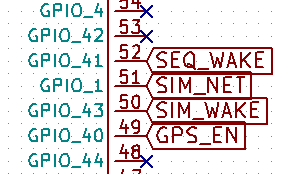


The sim card socket has a switch (SW1<->SW2) that will close (short) when the card is inserted. PD5 must be configured with a pullup resistor to detect the closed switch to ground. Therefore; SIMCARD, signalling PD5, uses negative logic. When the pin is high there is no sim card in the socket. For highest power efficiency the port-pullup should only be enabled several microseconds before a reading is taken, and disabled immediately afterward, else idle currents will continue to flow.

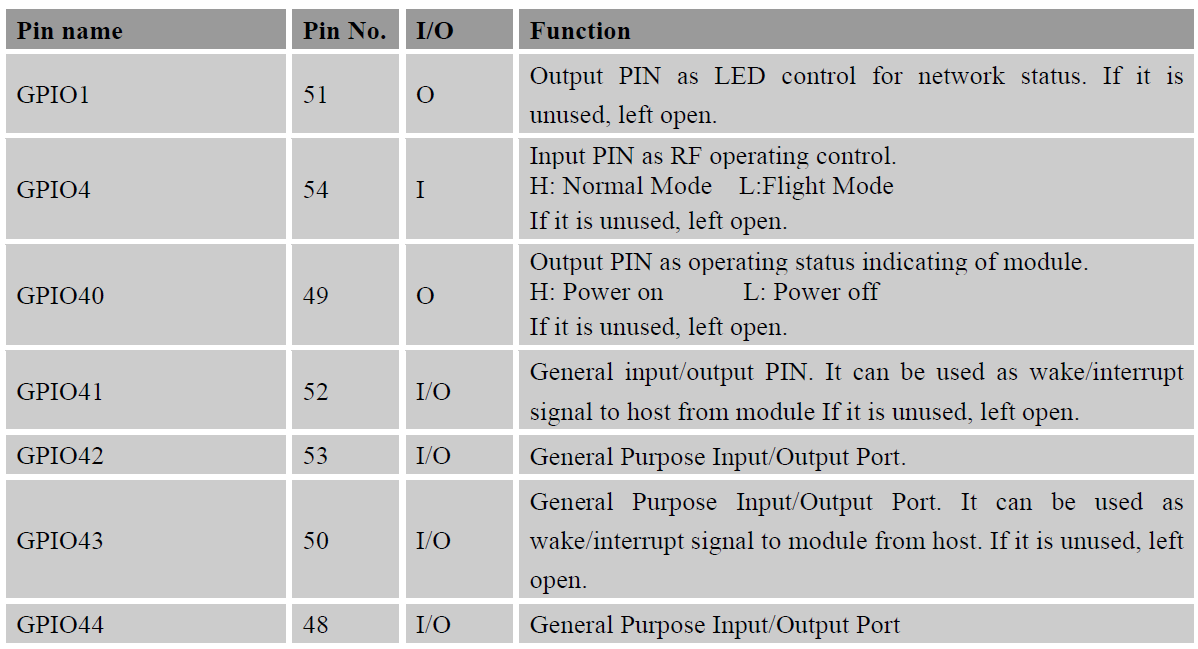
The SIM connections at the ATmega:



The GPIO connections at the SIM module are:



GPIO functions from the data sheet

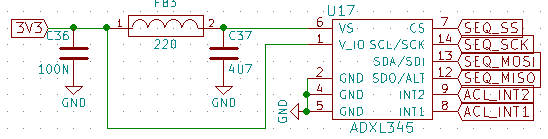


### GPS\_EN

The SIM5320 module is default configured to indicate its power state from GPIO40 (as seen above). The GPS\_EN signal sent by this pin serves two purposes; it allows the ATmega to directly sample the power state of the SIM, and it enables the active antenna power supply, necessary for the GPS antenna to operate.

## Accelerometer

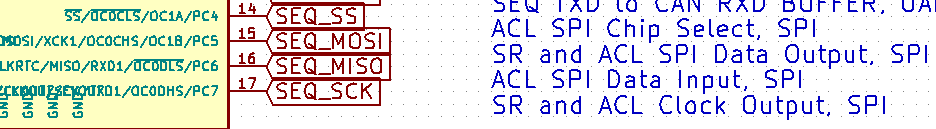
Accelerometer measurements are taken by an ADXL345 produced by analog devices. The device has 13 bit accuracy over a 16g range. It also has functions for automatically detecting orientation and user “taps”. The Atmel microcontroller accesses the accelerometer by the SPI bus. Several signals are shared with the shift registers.



The accelerometer has also been wired with 2 direct interrupt pins for waking the Atmel controller if a movement is too extreme. The interrupt pins can be configured by SPI command. Sample rates for accelerometer can range from 1/10 to 3200 Hertz sample rates.

The SPI signals have been tied to SPI peripheral pins on the Atmel. [ADXL345 Data sheet](http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf)





CANBUS Interface

The vehicle interface is provided by an STN1110, which is in many ways the same as an elm327. All three possible data networks have been provided ( CAN bus, j1850, ISO). The circuit has been optimized for low power usage. Many circuits that support operation are disabled while the STN1110 is in sleep or standby mode. CAN\_POWERDOWN is driven by the STN1110 to perform the optimize role.

All inputs from the obd2 connector have been protected from static shocks. Also, it is possible to measure the vehicle's battery voltage from a function built into the STN1110. Physical switch, SW1 provides a 120 ohm CAN bus Terminator resistor for Medium and high-speed networks.

To use the vehicle interface CAN\_RESET is taken high to bring the device out of reset. CAN\_SLEEP is also taken high to make the device active. At this point CAN\_TXD and CAN\_RXD are used to communicate exclusively with the STN1110.

[STN1110 Datasheets](http://www.scantool.net/scantool/downloads/97/stn1110-ds.pdf)

## CAN Start Up

The simple setup command set for the STN1110 for JLR vehicles is:

ate0,

ath1,

ats1,

atcaf0,

atspab

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