SUN DEVIL ROCKETRY

Avionics Hardware Design Team

Base Flight Computer Hardware Documentation

PCB Part No.	A0001
Current Revision	1.0
Project:	Base Flight Computer
Design Engineer(s)	Colton Acosta
Test Engineer(s)	Colton Acosta

TABLE OF CONTENTS

LISTING OF COMMON ACRONYMS

Acronym	Definition
AC	Alternating Current (Voltage)
ARM	Advanced RISC Machines, refers to processor architecture
ADC	Analog to Digital Converter, refers to MCU peripheral
DC	Direct Current
ESR	Equivalent Series Resistance (Capacitor/Inductor Specification)
GPIO	General Purpose Input Output, refers to MCU pin configuration
I2C	Inter-Integrated Circuit, serial communication protocol
IC	Integrated Circuit
IDC	Insulation-Displacement Contact, refers to cable header and harness assembly
IO	Input/Output
LED	Light-Emitting Diode
RTC	Real Time Clock, refers to MCU peripheral
MCU	Microcontroller

NC	Not connected
PC	Personal Computer (Not an embedded computer)
PCB	Printed Circuit Board
PWM	Pulse Width Modulation, refers to both the MCU peripheral and process of pulsing a voltage high and low at a predetermined rate and duty cycle
PWR	Power
RF	Radio Frequency
RGB	Red, Green, Blue
RMS	Root Mean Squared
SDR	Sun Devil Rocketry
SWD	Serial Wire Debug, refers to MCU programming and debug interface/pins
UART	Universal Asynchronous Receiver-Transmitter, refers to MCU peripheral
USB	Universal Serial Bus

PCB NAMING CONVENTIONS

All PCBs designed by Sun Devil Rocketry follow the same five character naming convention. Each board has its own unique part number which consists of a letter followed by a four digit number. The letter indicates the project focus, and the number indicates the order in which the boards were designed with lower numbers being assigned to older boards. The letter designations are listed in the table below.

Sun Devil Rocketry PCB Letter Designations

Letter	Definition
A	Avionics - General purpose boards to be used internally by the avionics team or non-propulsion avionics projects
H	Hybrids - Boards used in the projects of the hybrid propulsion team
L	Liquids - Boards used by SDR’s liquid engine
S	Solids - Boards used by the solid propulsion team’s projects

A complete listing of all Sun Devil Rocketry PCBs can be found on the Sun Devil Rocketry website.

PCB DESIGN FILES

The working directory for the A0002 design files can be found here for Sun Devil Rocketry members. The most up-to-date design files can also be downloaded by anyone on the Sun Devil Rocketry GitHub (link).

1. DESIGN OVERVIEW:

The SDR Avionics team aims to design and develop a general purpose flight computer to facilitate the acquisition of flight data and to serve as a prototyping platform for future projects in high powered and amateur rocketry. Not only will the project provide a convenient platform for students wanting to work on controls projects, but it will also help centralize the development of in demand skill sets such as electronics design, software engineering, and PCB fabrication within SDR such that the propulsion teams may spend more time efficiently working within the areas of engineering that are of interest to them. Requirements for the flight computer are somewhat arbitrarily derived from the basic functionality of currently available commercial altimeters such as the stratologger, which has been used in SDR high-powered flights previously. A few basic requirements for enabling more advanced control functionality have also been included in anticipation of future rocketry projects. The system Requirements for the computer are as follows:

SR01: The flight computer shall store important flight data in non-volatile memory via a purely electrical connection.

SR02: Flight data shall be accessible to a PC application through a direct USB connection between the computer and the flight computer (A0002).

SR03: The flight computer shall be capable of transmitting telemetry to a ground-station computer, even if the rocket is constructed with an RF-blocking material such as Carbon Fiber.

SR04: The flight computer shall support input voltages ranging 3.5 V to 15V for basic parachute deployment and data logging functionality.

SR05: The flight computer shall be capable of driving at least four servo motors with a PWM interface.

SR06: The flight computer shall be computationally capable of performing real time signal processing computations such as matrix inversions, FFTs, and eigenvalue problems.

SR07: The flight computer shall support dual parachute deployment. (drogue and main).

SR08: The flight computer shall be capable of differentiating between powered and unpowered flight.

SR09: The flight computer shall produce noise to indicate arming status prior to flight.

SR10: The flight computer shall conform to a form factor small enough to be mounted within the electronics bay of a level one high-powered rocket. A level one high-powered rocket may be assumed to have a minimum diameter of 2.6 in.

2. CONTROLLER ARCHITECTURE:

The high-level functionality of the flight computer hardware is defined by the block diagram shown in Fig. 2.1. The information and power relationships between each component/subsystem are denoted by the arrows with the indicated directionality.

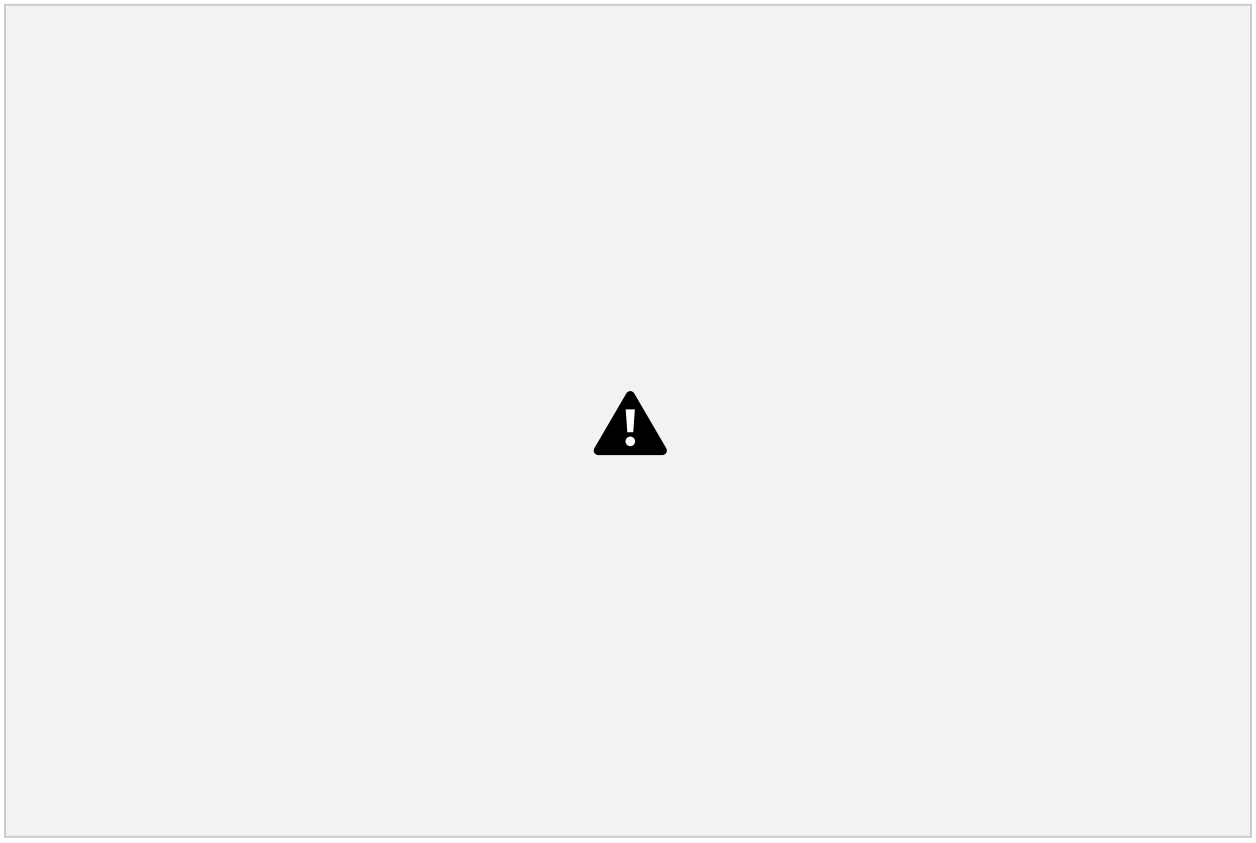


Figure 2.1 Flight Computer Architecture Diagram

High Level Requirements:

The high level requirements define the basic implementation details of the system requirements, and describe how the system architecture shown in Fig. 2.1 meets the system requirements. The system requirements met by each high level requirement are listed in parentheses. The high level requirements are as follows:

HR01: The flight computer shall store flight data in flash memory. (SR01)

HR02: The flight computer shall have a micro USB port. (SR02, SR10)

HR03: The flight computer MCU shall be powered with a 3.3V voltage regulator (SR04)

HR04: The flight computer shall have two power connectors, and allow independent supplies to be connected simultaneously (SR05).

HR05: The flight computer shall include six screw terminals for ematch and switch wires. (SR07)

HR06: The flight computer MCU GPIO should be used to detect ematch and switch continuity. (SR07)

HR07: The flight computer shall have an embedded IMU. (SR09)

HR08: The flight computer shall indicate arm and continuity status with a buzzer. (SR09)

HR09: The flight computer shall be no larger than 2 in wide. (SR10)

HR10: The flight computer shall include a barometric pressure sensor. (SR07)

3. POWER SUPPLY:

3.3V Supply:

The engine controller's microcontroller requires a 3.3V supply to operate, which is delivered from a linear 3.3V regulator powered by the 5V supply rail. To allow the regulator to be supplied from several 5V sources, a power multiplexor is used that automatically switches between 5V sources based on whether or not power is present on the rail. The multiplexor and linear regulator application schematic is shown in Fig. 3.3.

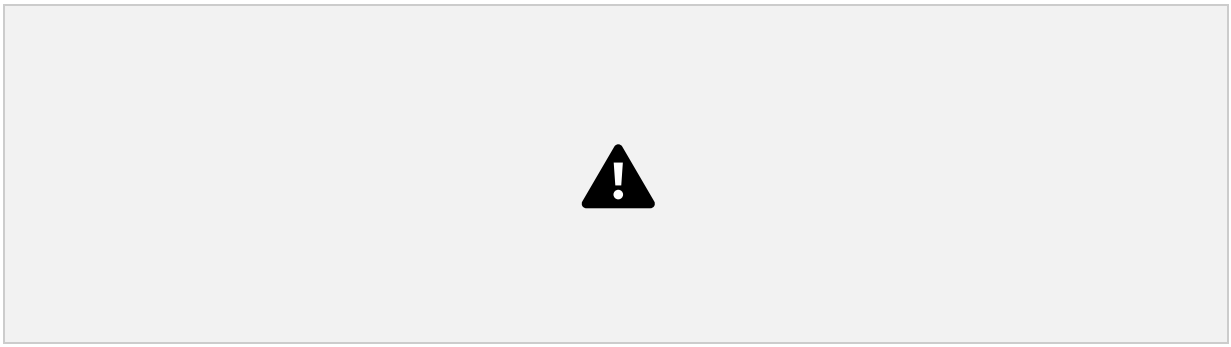


Figure 3.3 Power Multiplexor and 3.3V Linear Regulator

An indication LED is used on the 3.3V power line to visually indicate when power is on the rail. The current limiting resistor was chosen to limit the LED current to approximately 5mA. The regulator can supply up to 1.2A. Two 10uF tantalum capacitors are used at the regulator input and output pins to enhance the regulator's stability and transient response, as recommended in the device datasheet.

The power multiplexer supplies the 3.3V regulator using either the 5V output of the buck converter or the 5V USB rail. The D0 and D1 pins set the switching mode of the multiplexor. Setting D0 disconnected and D1 grounded puts the multiplexor in auto-switching mode. In this mode, the multiplexer passes the IN1 voltage to the output by default. If the IN1 voltage falls below ~2.5V, the multiplexor will pass the IN2 voltage to the output if it is greater than ~2.5V. This effectively prioritizes the buck converter power supply over the USB rail, to allow the USB supply current to be minimized when large current draw is required by the controller. The USB supply is mainly for use when communicating with a PC, to make software development more convenient by limiting the amount of external circuitry required to power the board. Both input voltage pins are decoupled using 0.1 uF ceramic capacitors placed close to the chip in the PCB layout. This suppresses noise and inductive spikes on the input rails that can influence the multiplexor switching or damage the device.

The ILIM pin controls the current-limiting capability of the multiplexor. The relationship between the maximum output current and the pull-down resistance is given in pg. 17 of the datasheet. The resistor chosen in this application was chosen arbitrarily to make the current limit close to the maximum current of the device, as the datasheet (pg. 17) advises against disabling the current-limiting feature by shorting the ILIM pin to ground. The STAT pin indicates which supply voltage is passed to the multiplexer output. The multiplexor grounds the STAT pin when the IN1 voltage is passed to the output, and floats the STAT pin when IN2 is used. The pull-up resistor on the STAT pin ensures that the 5V_SRC signal goes high when the IN2 supply is used.

Microcontroller Power Supply Scheme:

In order to ensure the MCU receives a stable voltage from the regulator circuit despite the presence of parasitics present in the PCB routing, a number of ceramic, low-ESR, decoupling capacitors are used as recommended by the datasheet. In the PCB layout design, they are placed as close to the MCU pins as possible in order to minimize the trace lengths. The number of decoupling capacitors and the capacitance of each were chosen according to the datasheet recommendations. The recommended connections and capacitances are shown in Fig. 3.4. These capacitors are grouped together on the schematic to save whitespace, and are shown in Fig. 3.5.

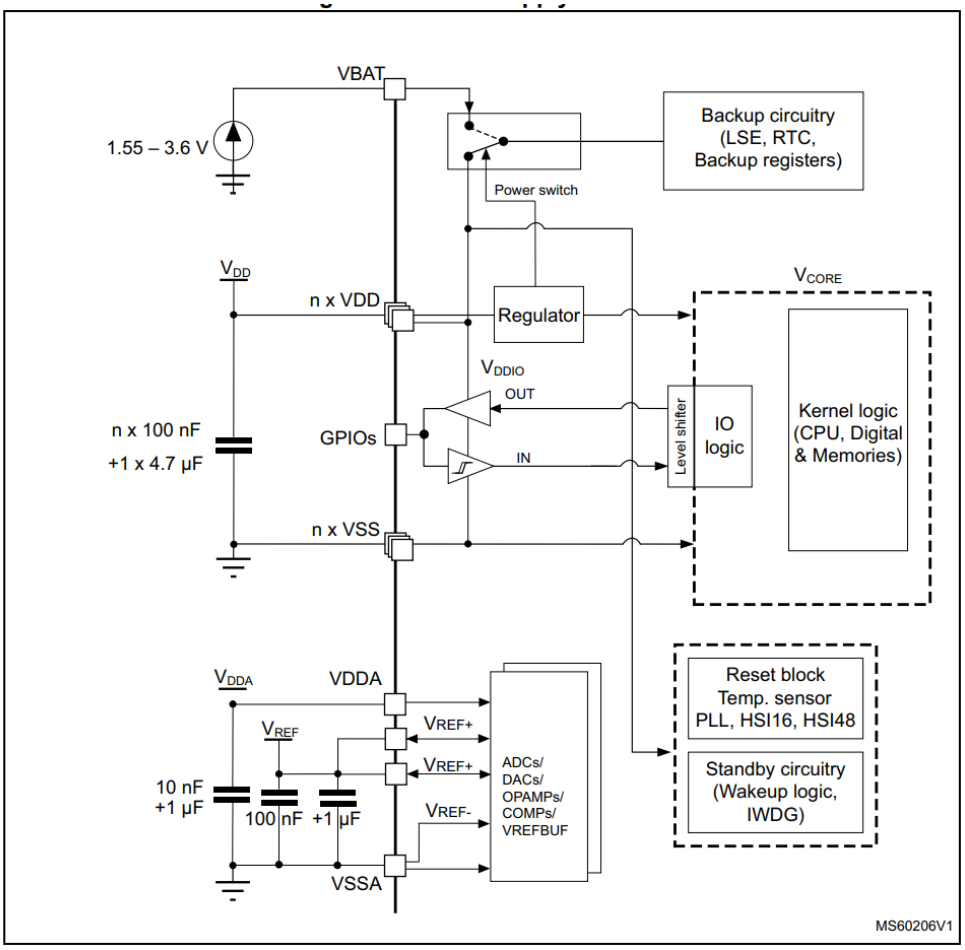


Figure 3.4 MCU Power Supply Scheme (Datasheet 5.1.6, pg. 71)

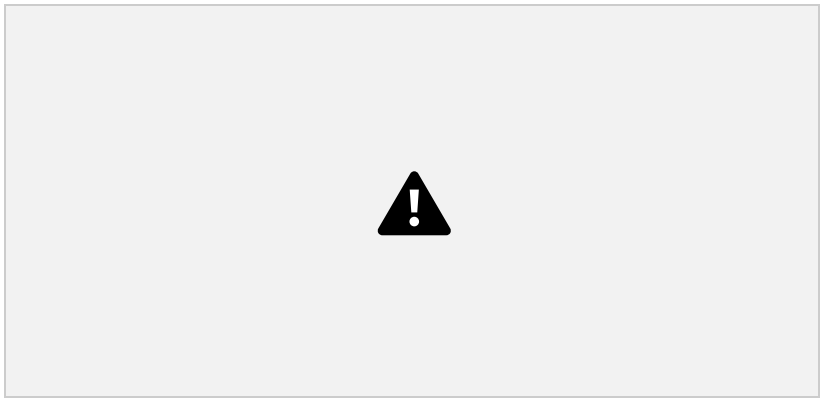


Figure 3.5. Microcontroller Decoupling Capacitors in Schematic

The USB and VDDLDO pins shown in Fig. 3.4 are not included with the chosen MCU, and therefore these capacitors are left out of the schematic. Additionally, the internal regulator of the MCU requires external 2.2 uF low ESR ceramic capacitors for stability (Datasheet 6.3.2 pg. 96). These capacitors are grouped separately from the decoupling capacitors to distinguish them from the other MCU capacitors, although they are also placed very close to the MCU in the PCB layout. The MCU stability capacitors' schematic are shown in Fig. 3.6.

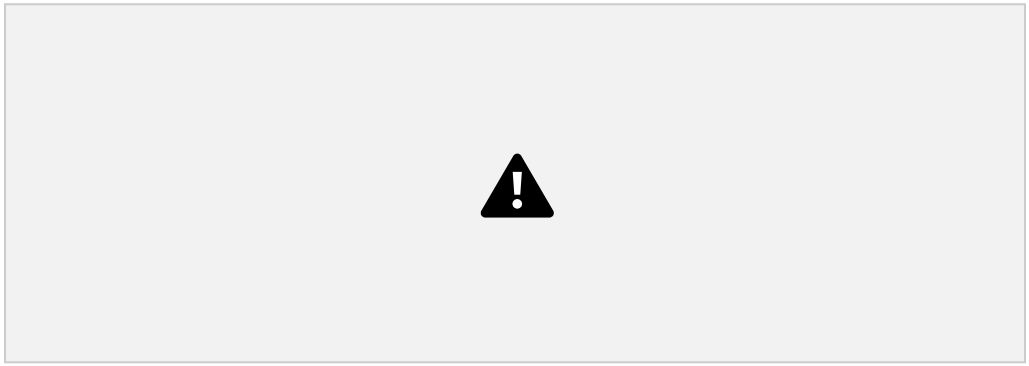


Figure 3.6. MCU Regulator Stabilization Capacitors

4. MICROCONTROLLER:

The engine controller uses an STMicroelectronics STM32H750VBT6 microcontroller, with a single-core ARM Cortex-M7 processor. The MCU was chosen for its maximum clock speed of 480 MHz, since the timing of the engine sequencing is a critical performance factor that influences design decisions. Although the MCU has limited availability and a high per-unit price, the thorough documentation available offsets these costs. The peripheral usage of the MCU is given in Table 4.1.

Table 4.1 Microcontroller Peripheral Functionality

Peripheral	**Usage**
Pulse Width Modulation (PWM)	Sets the color of the firmware status LED
Timer	Triggers the execution of precise engine sequencing operations
Analog to Digital Converter (ADC)	Reads pressure transducer, load cell, and strain gauge sensor data
Universal Asynchronous Receiver Transmitter (UART)	Valve controller (L0005) serial communication, USB PC communication
Inter-Integrated Circuit (I2C)	Communication with thermocouple cold-junction compensation chip
Serial Peripheral Interface (SPI)	Data logging on flash and SD card external memory devices, Wireless Interface
General Purpose Input/Output (GPIO)	Ignition, power indication, amplifier gain selection

Programmer:

The MCU is programmed using an SWD interface and the standard 20 pin ARM IDC programming cables. The programmer used is the ST-Link V2 (source), chosen for its affordability and compatibility with ST microcontrollers. A 0.05” (1.27mm) pitch mating IDC connector is used for its small form factor, which requires an adapter (A0004) to connect to the 0.1” (2.54mm) IDC header on the ST-Link.

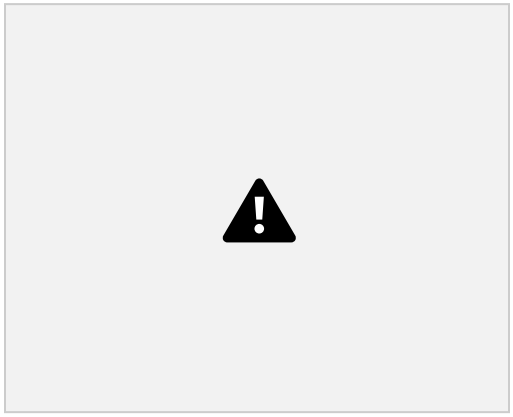


Figure 4.1 ST-Link V2 Programmer

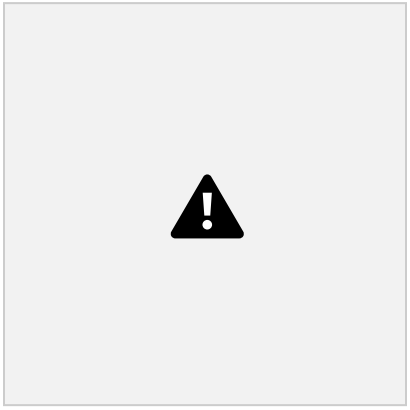


Figure 4.2 IDC Programming Cable

The SWD pinout of the ST-Link V2 is shown in Figure 4.3, which is taken from the ST-Link datasheet (pg. 13). The “Not Connected” pins are used for JTAG, and therefore are not needed for this application. The ST-Link pins used during SWD programming and debugging are listed in Table 4.2.

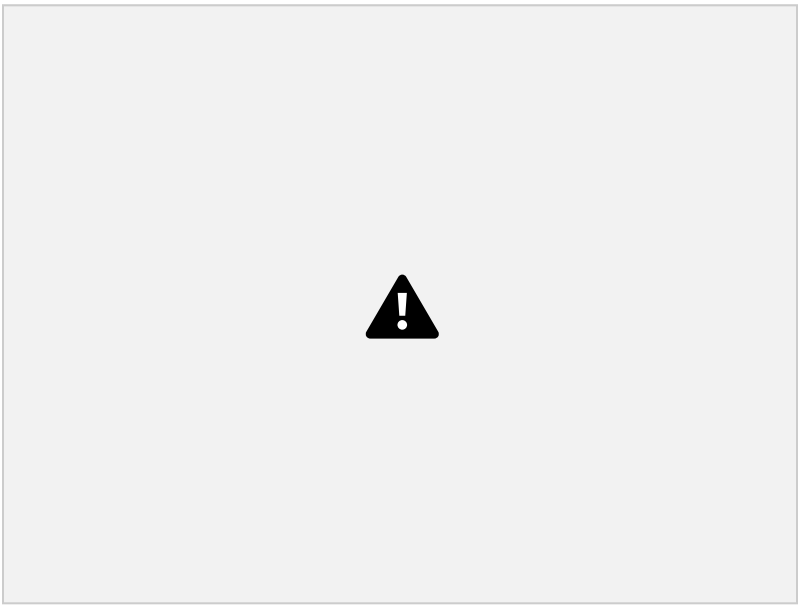


Figure 4.3. ST-Link V2 Programmer Pinout

Table 4.2. ST-Link V2 SWD Pinout Descriptions (ST-Link datasheet, pg. 12)

Pin Number	Name	Description
1-2	MCU VDD	Target reference voltage, connected to MCU supply voltage (3.3V)
7	SWDIO	Bi-directional data, pulled up to 3.3V
9	SWCLK	Clock signal, pulled up to 3.3V
13	SWO	Serial Wire Output trace port
15	NRST	Reset signal, active low

The programmer schematic is shown in Fig. 4.4. The SWDIO and SWCLK signals use 100 kOhm pull up resistors as recommended by the MCU reference manual (pg. 3061).

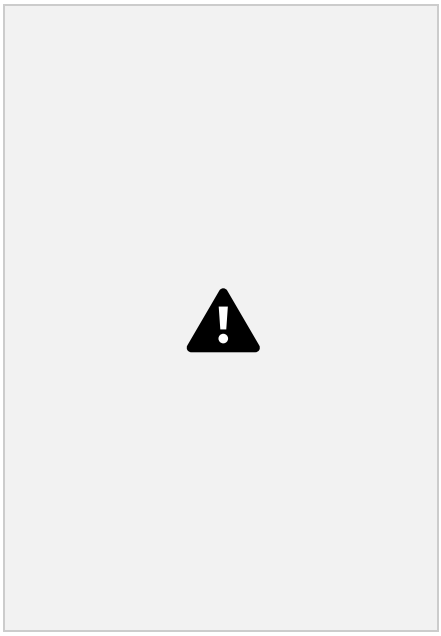


Figure 4.4 SWD Programming Connector Schematic

Microcontroller Pinout:

The MCU pinout provided in the MCU datasheet (pg. 53) is shown in Fig. 4.5.

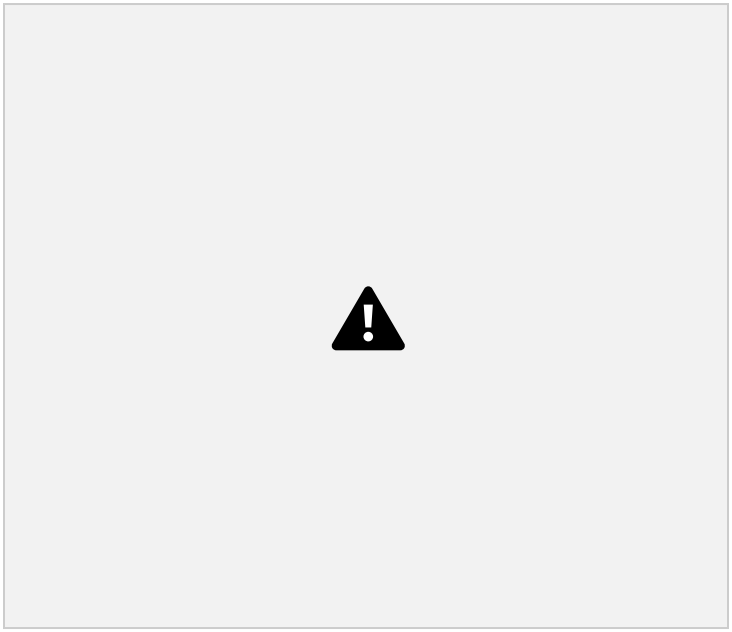


Figure 4.5 Microcontroller Pinout (MCU datasheet, pg. 53)

To improve the readability of the MCU schematic and PCB layout, each MCU pin used in the design is attached to a global net label. The labels used are shown in the MCU schematic shown in Fig. 4.6, and are listed in Table 4.3. Enumerated signals in Table 4.3 use lowercase “n” to

indicate the signal number. The signals detailed in Table 4.3 can also be found in a google sheets “cheat sheet” in the L0002 design folder for more convenient access and reference.

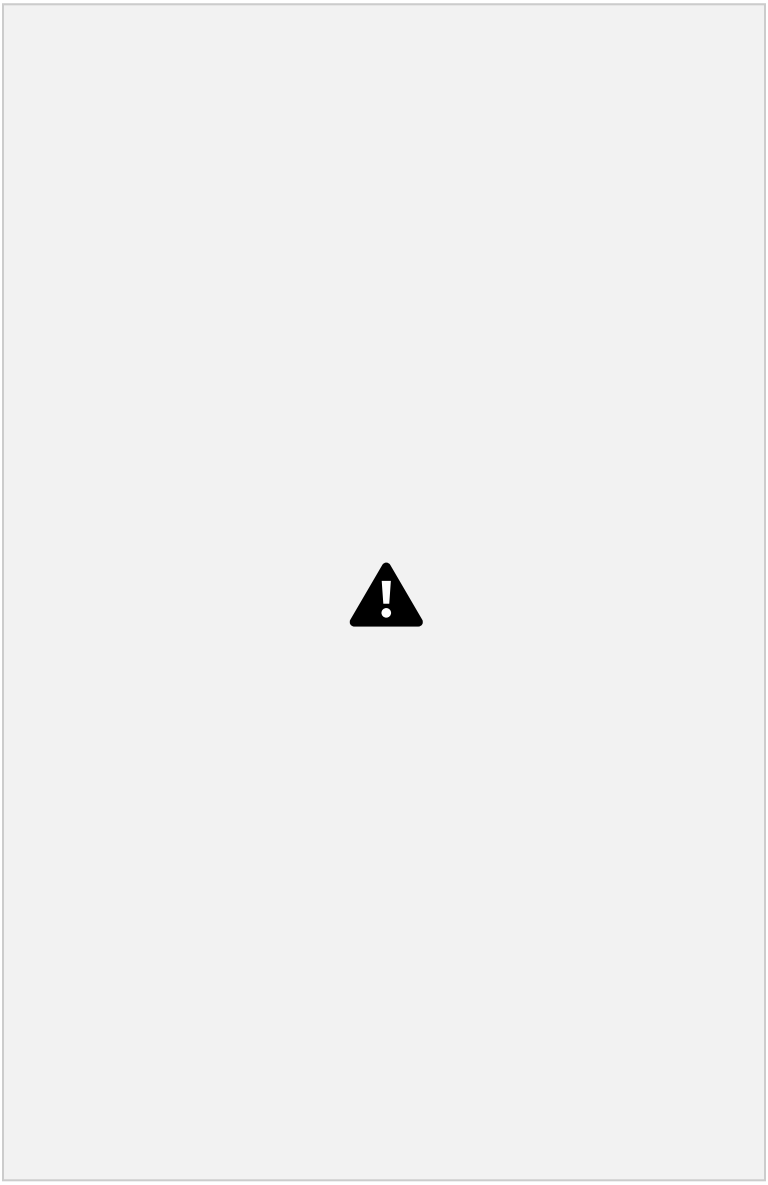


Figure 4.5. Microcontroller Schematic

Table 4.3 Microcontroller Signal Names and Definitions

Signal Name	**Signal Type**	**Definition**
5V_SRC	Digital Input	Indicates which 5V power supply is being used by the multiplexer.
BOOT	Digital Input	MCU boot mode, high signal prompts MCU to run code from boot loader memory instead of main

		program flash
CAPn	Power	Internal regulator stability capacitors. See “Microcontroller Power Supply Scheme” in section 3 for more details. n = 1 or 2
E_CONT	Digital Input	Indicates Ematch Continuity. High for continuity, grounded when ematch is disconnected
FIRE	Digital Output	Ignites Ematch to ignite Solid Propellant Slug. 3.3V Signal pushes current through power MOSFET to Ematch
FLASH_HOLD	Digital Output	Flash HOLD mode signal
FLASH_MISO	Serial Communication	Data logger SPI master (MCU) input, slave output line
FLASH_MOSI	Serial Communication	Data loggerSPI master (MCU) output, slave input line
FLASH_SCK	Serial Communication	Data logger SPI clock line
FLASH_SS	Serial Communication	Flash chip slave select line (SPI network)
FLASH_WP	Digital Output	Flash write protect
NOZ_CONT	Digital Input	Indicates Continuity of Nozzle Wire. Gets pulled high when nozzle exhaust burns through wire
NRST	Digital Input	MCU reset, active low
RX_USB	Serial Communication	UART to USB Serial Transmission, MCU Transmit, USB Host Receive
SD_DETECT	Digital Input	Indicates when an SD card is inserted in the SD card jack. Pulled low when the SD card is inserted.
SD_SS	Serial Communication	SD card slave select line (SPI network)
SP_CONT	Digital Input	Indicates Continuity of solid propellant wire. Gets pulled high when solid propellant burns through wire

STATUS_R	Digital Output	Firmware status LED. Controls the red value of the status LED
STATUS_G	Digital Output	Firmware status LED. Controls the green value of the status LED
STATUS_B	Digital Output	Firmware status LED. Controls the blue value of the status LED
SWCLK	Serial Communication	SWD programming clock signal, generated by programmer (ST-Link)
SWDIO	Serial Communication	SWD programming data line
SWO	Serial Communication	SWD serial trace signal
TX_USB	Serial Communication	UART to USB Serial Transmission, MCU Receive, USB Transmit
~USB_RST	Digital Output	Resets the USB transceiver, active low, open drain
~USB_SUSPEND	Digital Input	Driven low by USB transceiver when controller enters the suspend state
ValveCtrl_RX	Serial Communication	Valve control UART receiver. Receives UART responses from the valve controller (L0005)
ValveCtrl_TX	Serial Communication	Valve control UART transmitter. Issues valve control commands to be executed by the valve controller (L0005)

LED Indication:

The status of the MCU firmware is visually displayed using the status LED and signals. The status LED is an RGB LED controlled by PWM signals that allow the LED to display any arbitrary RGB color. The status LED schematic is shown in Fig. 4.6.

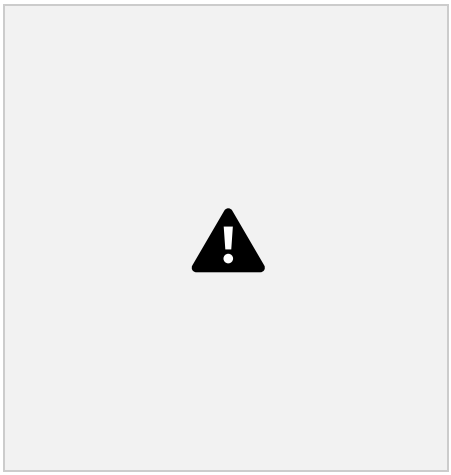


Figure 4.6. Firmware Status LED Schematic

Microcontroller Reset:

The microcontroller reset pin is driven by the SWD programmer and a tactile button that allows the MCU to be reset without removing the power supply. A 0.1 uF ceramic capacitor is used to prevent unwanted MCU resets due to voltage spikes on the reset pin. The button is disabled by default, and is enabled by closing the jumper with a solder bridge. The reset circuit is shown in Fig. 4.7.

Figure 4.7. Microcontroller Reset Button

USB Interface:

The engine controller uses a USB interface to directly communicate with PCs. The USB interface allows for easy and unrestricted access to the engine controller state and hardware. Since the chosen MCU does not include a USB interface, an external USB transceiver is used which converts UART serial data from the MCU to the USB protocol. The transmitted data is then read by the PC by monitoring the relevant serial port. The USB transceiver schematic is shown in Fig. 4.8.

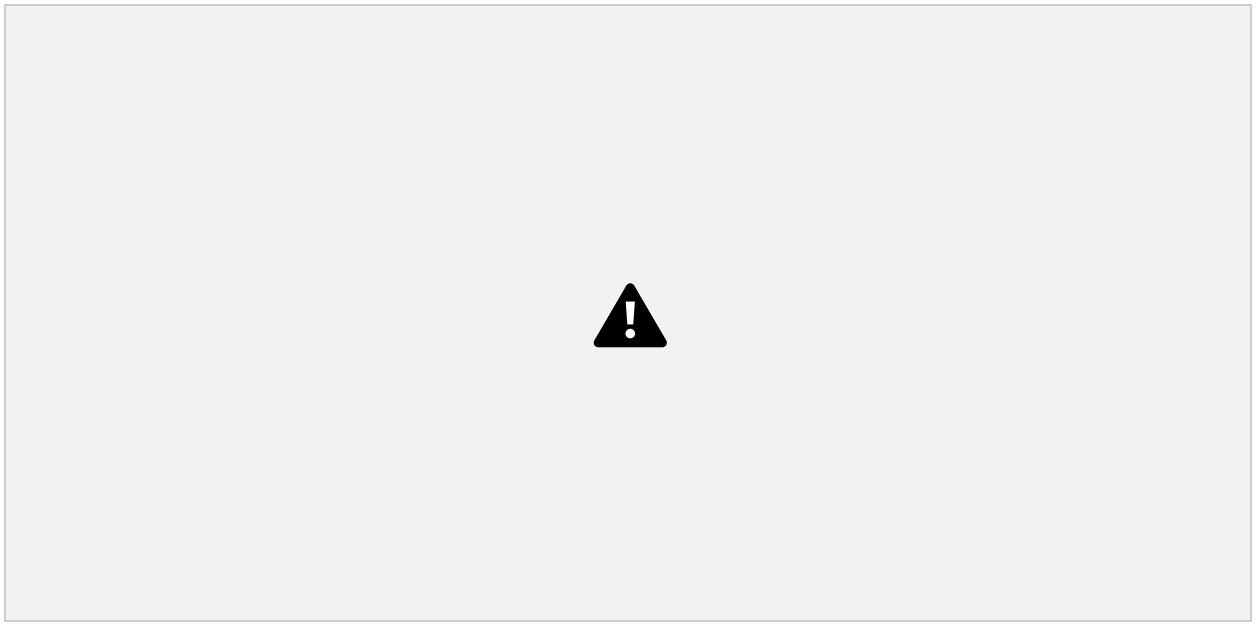


Figure 4.8. UART to USB Converter Schematic

The transceiver is powered by the USB 5V rail, and the transceiver's internal regulator is decoupled using two ceramic capacitors with values recommended by the transceiver datasheet (pg. 14). The transceiver is compliant with the USB 2.0 specification, and transmits/receives USB packets at a data rate of 12 Mbps. The RST and SUSPEND pins are used to control/monitor the behavior of the USB transceiver. A low voltage to RST will reset the transceiver, and the pin is pulled up to 3.3V by a 4.7 kOhm resistor to increase noise immunity as recommended by the transceiver datasheet (pg. 14). The SUSPEND pin is driven low when the USB transceiver enters suspend mode. The suspend pin is pulled down to ground with a 10 kOhm resistor to prevent the suspend signal from being asserted when the transceiver is reset (datasheet pg. 14). The transceiver is protected from ESD by the zener diode array connected to the USB signals.

To ensure the USB signals are not corrupted by reflections on the data transmission lines, the USB D+ and D- traces are designed to have a characteristic differential impedance of 90 Ohms at the transmission frequency of 12 MHz. The trace specifications were calculated using KiCad's impedance calculator. Since the PCB dielectric is FR-4, the dielectric constant is 4.5, and the loss tangent is 0.02. The traces are copper, so the resistivity is 1.72e-8 Ohms/m. The dielectric height is calculated from the PCB thickness and number of layers to be 0.533 mm, since the controller is 1.6 mm thick and has 4 layers. The spacing between the differential traces was chosen to be 3.5 mil, since this is the minimum trace thickness that is supported by JLCPCB. The trace widths were then calculated to make the trace differential impedance as close to 90 Ohms as possible. A screenshot of the KiCad impedance calculator is shown in Fig. 4.9.

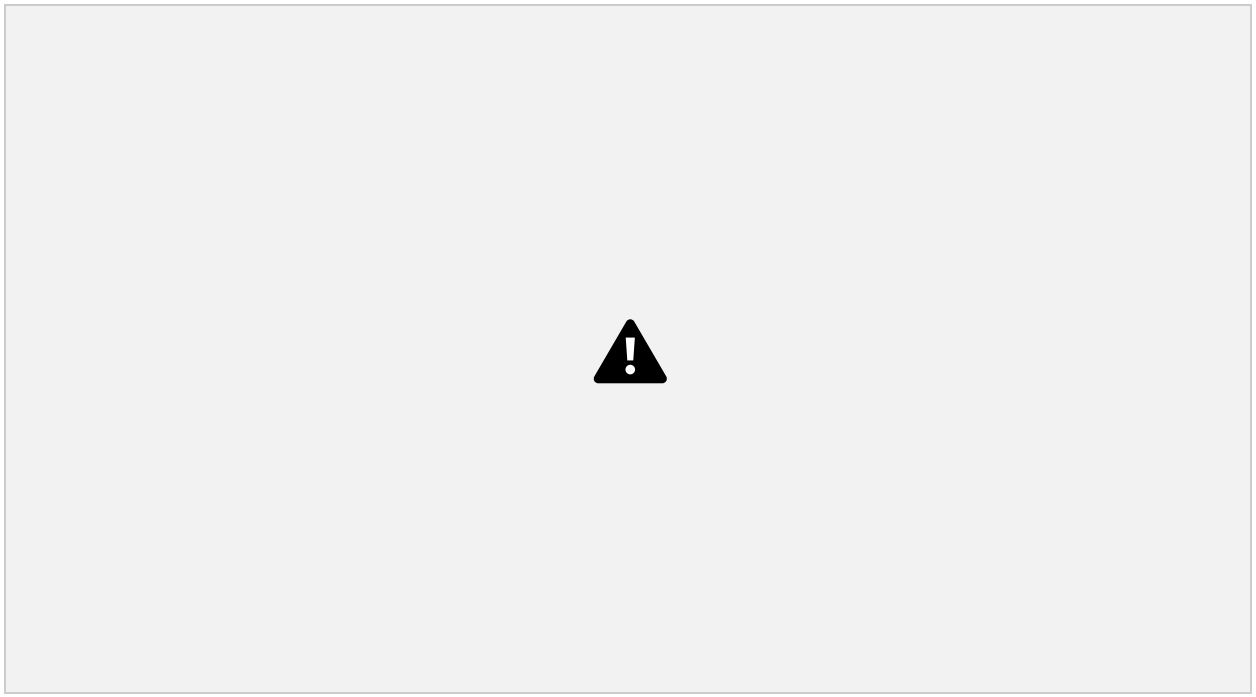
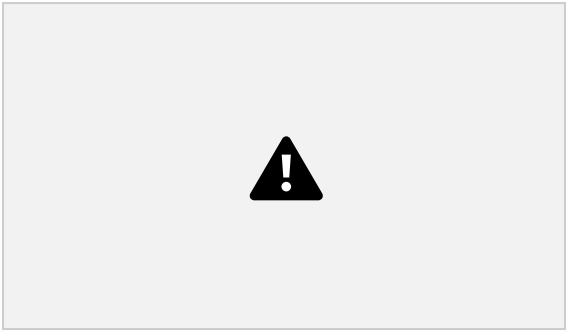


Figure 4.9. USB Transmission Line Impedance Calculations

External Oscillator:

MCU Reference Manual pg. 336



MCU Datasheet pg. 28: Crystal oscillator of 4-48 MHz

Crystal increases the accuracy of the clock and hence the timers.

Load Capacitance: 18pF (Crystal Datasheet, with part number)

Parasitic MCU capacitance: Datasheet recommends 10 pF

5. DATA LOGGER:

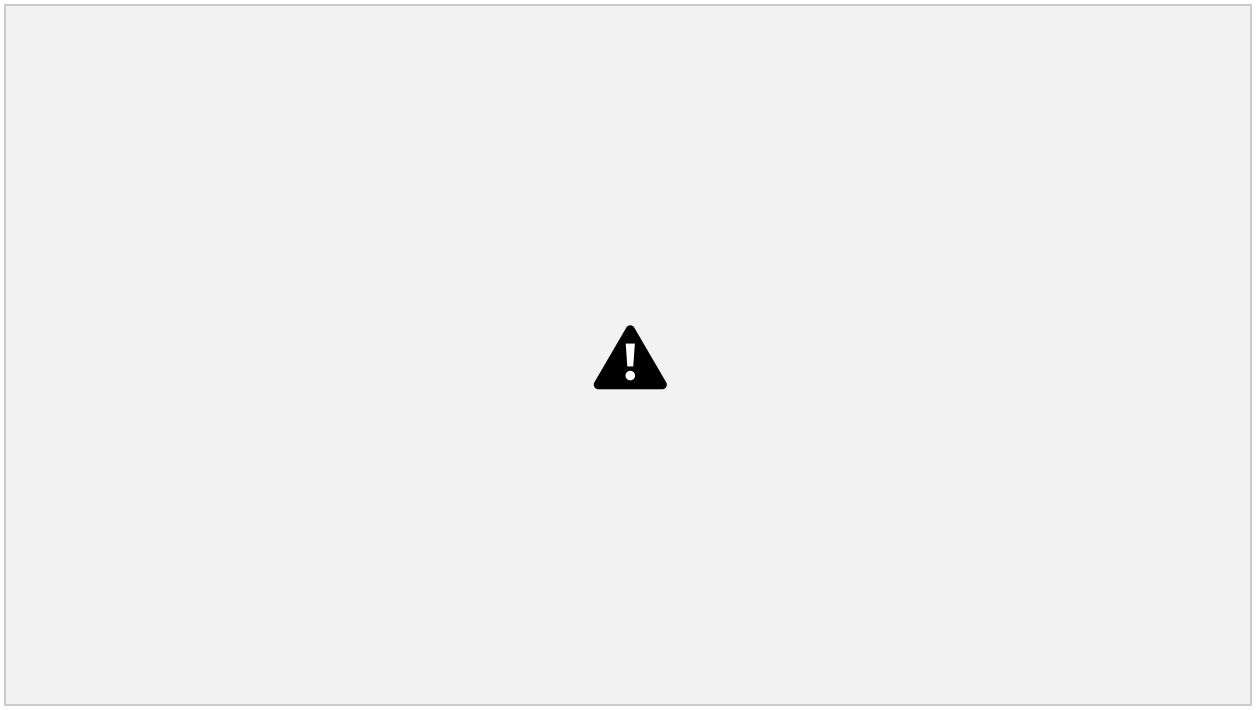
External Flash:

- Active low slave select signal, high to low transition on CE pin activates the chip
- 8 bit data transfers for opcode, address, and data
- Inputs accepted on rising edge of the CLK
- Data transfer starts with MSB (most significant bit)
- Chip supports either active high or active low clk signal
- Low idle clock is used with 10k pull down resistor (MCU CPOL bit 0)
- Rising edge of clock with low idle clock makes MCU CPHA bit 0 (1 edge)
- Open drain slave select pin to make slave select default high

6. TELEMETRY:

LoRa Module:

- 868 MHz up-conversion frequency
- RFM95 Chip
- Configuration registers must only be written in sleep or standby mode (Datasheet pg. 29)
- Operating modes - Datasheet pg. 31
- Implementation based on reference design (Datasheet pg. 119)
- 50 Ohm transmission line impedance to antenna (Datasheet pg. 13)



7. SENSORS:

IMU:

- VDD Bypass Capacitor: 0.1uF, 10% (MPU-9250 Datasheet pg. 20)
- VDDIO Bypass Capacitor: 0.01uF, 10% (MPU-9250 Datasheet pg. 20)
- Regulator Filter Capacitor (REGOUT pin), 0.1uF, 10% (MPU-9250 Datasheet pg. 20)
- VDDIO pin sets the MPU-9250 logic levels
- 10kOhm I2C pull ups to match MPU-9250 breakout board
- Auxiliary I2C interface not used since MPU-9250 has a magnetometer. Auxiliary I2C interface used to communicate with external accelerometers/gyros/magnetometers
- Reserved pin connections laid out on MPU-9250 datasheet pg. 19
- FSYNC pin connected to ground as it is unused (MPU-9250 datasheet pg. 19)
- AD0 pin connected to ground to set the LSB of the I2C address (MPU-9250 datasheet pg. 19)

GPS:

- EXTINT Pin: External Interrupt, Receiver control or aiding, connect to MCU PWM pin, can be programmed as open circuit protection
- TIMEPULSE: Outputs pulses synchronized with GNSS or UTC time grid, 1 pulse per second

- UART Interface not used, but signals are kept for potential future use
- I2C interface used with IMU
- Coin battery used as backup supply (Regulator needed?)
- LNA_EN: Used to turn on or off an external LNA or active antenna. Depends on antenna so not used for now.
- VCC_RF: Supplies power to an active antenna, Depends on antenna so not used for now
- Max VCC and Backup Supply voltage: 3.6V (MAX-8 Datasheet pg. 15)
- Common Battery voltages are 3V, 3.7V, 1.2V and 1.5V
- 6.8mm coin battery for a small form factor (battery dimensions shown on datasheet pg. 12)
- All I/O pins use internal pull-ups, so no external pull-ups required (HW Integration manual pg. 8)
- Add protection to IO lines to prevent EMI from damaging the board (HW Integration manual pg. 9). Datasheet advises use of series resistors, inductors, or ferrite beads. Series resistors are used for convenience. The HW integration manual recommends a resistance greater than 20 Ohms. 33 Ohms is used as a good compromise between EMI protection, availability, price, and fast rise times.
- UART pins may be configured as TX ready for DDC interface (I2C, HW Integration manual pg. 11)
- Potentially use an external LNA if a passive antenna is used
- Antenna must have sufficient bandwidth to receive GNSS constellations
- For active antenna design, see the hardware integration manual pg. 15

Barometric Pressure Sensor:

- 0.1uF decoupling caps for VDD and VDDIO supply pins as recommended by datasheet (pg. 48)
- 4.7kΩ I2C pull-up resistors, recommended by datasheet (pg. 48)
- INT: BMP interrupt pin
- CSB: Chip select, not used by I2C interface, datasheet recommends to connect to VDDIO (pg. 48)

8. PARACHUTE DEPLOYMENT:

The ignition system is responsible for pushing at least 2A of current through the ignition ematch once the FIRE signal is asserted by the MCU. Additionally, the controller provides three continuity readings to provide the controller feedback on the ignition state of the engine. These ports are to be used to indicate ematch continuity, solid propellant continuity, and nozzle continuity. In order to supply 2A of current to the ematch using the low-power MCU signal, a power MOSFET is used to sink the ignition current. Since the 5V supply cannot supply 2A of current, the ematch current is drawn from the input voltage, which may be any voltage from 5V

to 42V. For practical reasons, the assumed supply voltage is 12V since this is a reasonably low but common supply voltage. Since the supply voltage is somewhat arbitrary, a comparator is used to generate the e-match continuity signal. The ignition schematic is shown in Fig. 9.1, and the additional continuity schematics are shown in Fig. 9.2.

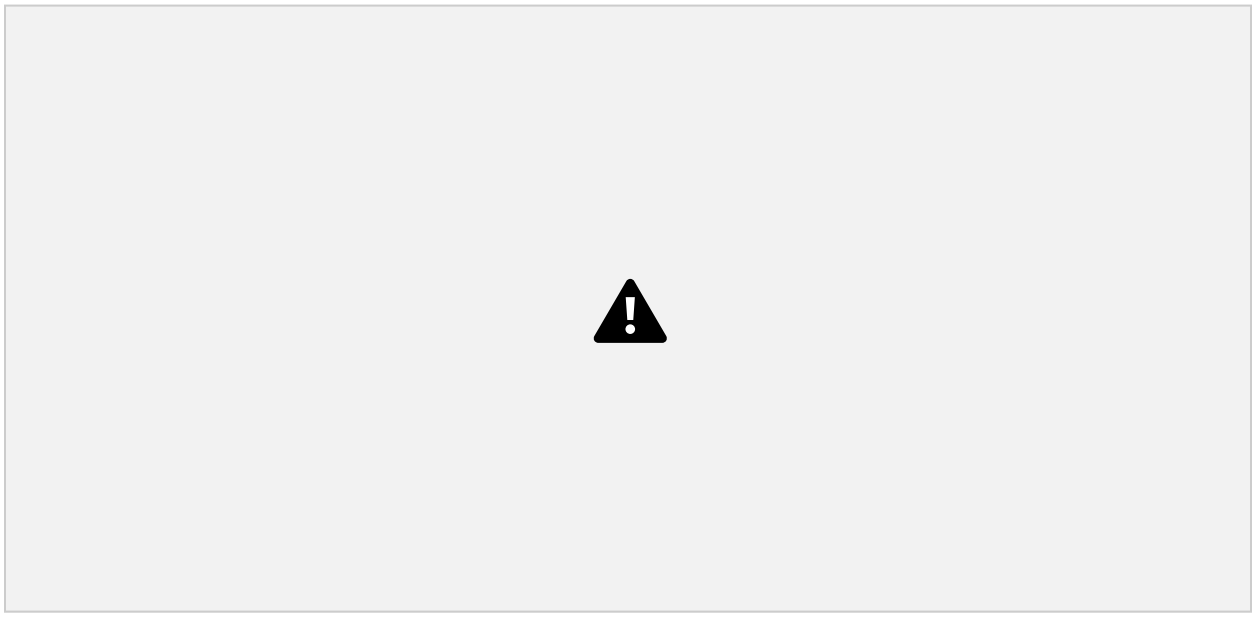


Figure 9.1 Ignition and E-Match Continuity Schematic

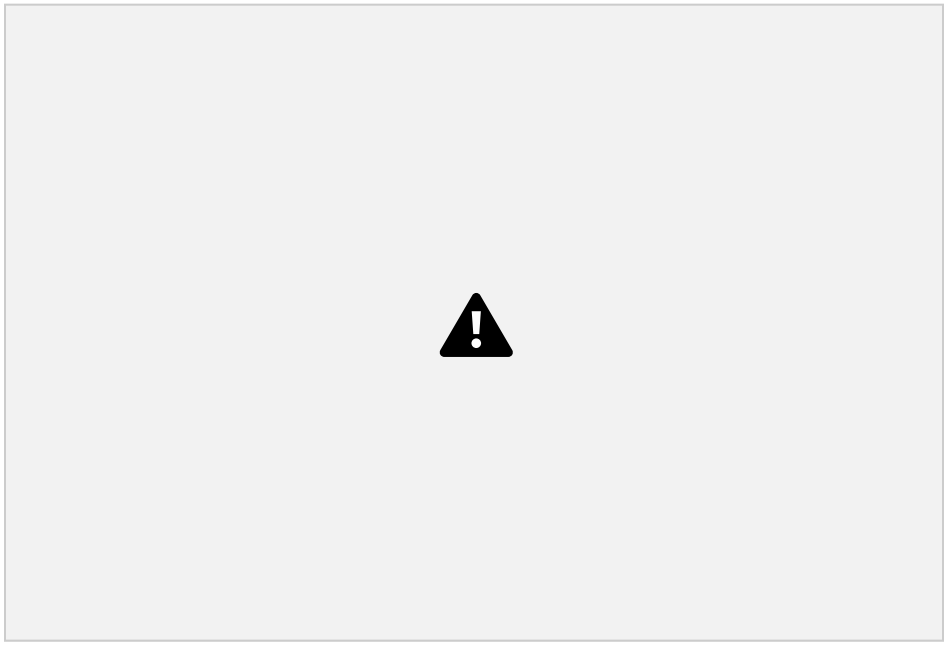


Figure 9.2. Continuity Wire Schematic

Power MOSFET:

The chosen MOSFET to drive the e-match current (Q1 in Fig. 9.1) has a maximum continuous current rating of 2A and a maximum pulsed current rating of 8A. Therefore, this MOSFET should be capable of providing the needed current for the short duration of time needed to ignite the e-match. The MOSFET has a maximum drain-to-source voltage of 30V, which ensures that the e-match can be used with input voltages up to 30V. To demonstrate the capacity of the MOSFET to supply sufficient current when a 3.3V signal is applied to the gate by the MCU, the current-voltage characteristic of the MOSFET is shown in Fig. 9.3

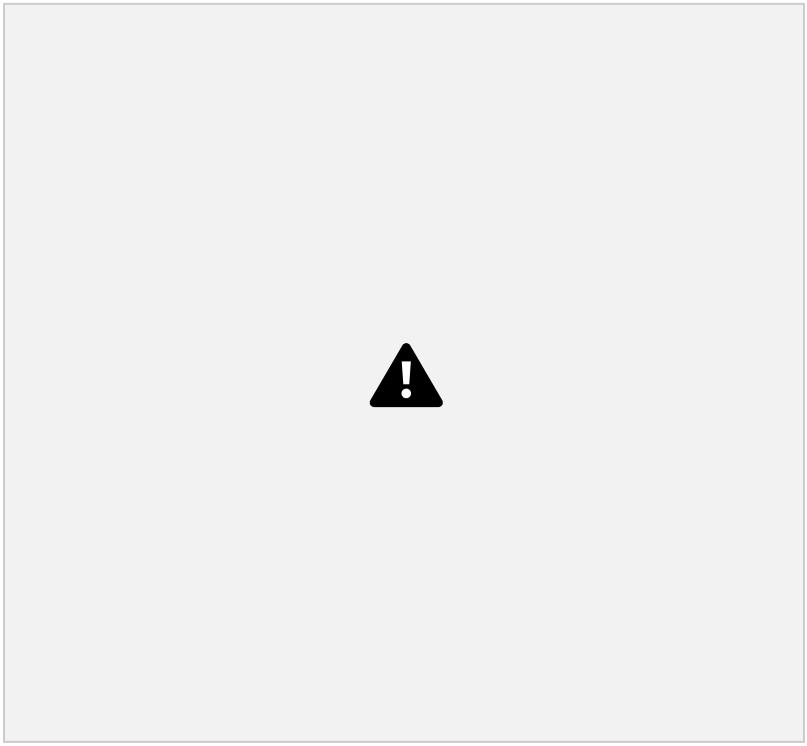


Figure 9.3. Power MOSFET Current-Voltage Characteristic (Datasheet pg. 3)

In Fig. 9.3. it can be seen that when 2.0V is applied to the MOSFET gate 2 amps are sourced when the drain-to-source voltage is 10V. Therefore, a 2 amp current surge through the e-match when 3.3V is applied to the MOSFET gate is reasonable.

It can be seen in Fig. 9.1 that a 100kΩ resistor and 1μF capacitor are also connected to the MOSFET gate. These components are used to prevent the MOSFET from igniting the e-match without a push-pull input from the MCU. The resistor pulls the gate to ground in order to prevent a floating gate triggering the e-match, and the capacitor prevents high-frequency noise or surges from triggering the e-match.

Continuity Wires:

As shown in Fig. 9.2, the connectors P2B and P2C allow a continuity wire to be run through the engine to provide feedback to the controller on the combustion status. The “solid propellant continuity” port is to be used with a wire running through the solid propellant in the combustion chamber such that the wire continuity is disrupted once the solid propellant is burnt by the e-match. The “nozzle continuity” port is to be used with a wire run across the exit of the combustion chamber nozzle such that the wire continuity is disrupted when hot exhaust is expelled out of the nozzle. Before continuity is broken during ignition, the continuity wires connect the MCU pins to ground. Once the connection is severed, the 10kΩ pull-up resistors pull the MCU pin to 3.3V. Thus, a high digital input at the MCU indicates a lack of continuity.

Comparator:

Since the input voltage powering the board may vary from 30V to 5V, continuity in the e-match is detected with the use of a comparator instead of a pull-up resistor. The comparator configuration is shown by U11 in Fig. 9.1. When the e-match is connected to the terminals of P2A, the non-inverting input of the comparator is connected to the input voltage. The inverting voltage is connected to the output of the 5V supply, which ensures that the comparator only indicates that the non-inverting input is greater than the inverting input when there is continuity in the e-match terminals. A 100kΩ pull-down resistor ensures that the non-inverting input is never left floating in order to prevent static charge on the screw-terminal from triggering the continuity indication. The chosen comparator uses an open-collector output, which allows the comparator to output a 3.3V signal while being powered from the board input voltage. The 10kΩ resistor on the comparator output pulls the continuity MCU pin to 3.3V when there is no continuity in the e-match. The pull-up resistance choice was made arbitrarily, and to match other resistors on the board.

9. BUZZER:

- Connect buzzer to PWM of MCU
- Control frequency of buzzer sound with frequency of PWM
- 50% duty cycle PWM

10. FPGA:

11. LAYOUT CONSIDERATIONS:

12. REVISION HISTORY:

12.1 Revision 1.0