KREPE Flight Computer Hardware Manual

Matt Ruffner

July 10, 2020

Contents

List of Figures

List of Tables

1 Introduction

This document has pin names and connections, along with implementation notes and design choice explanations. Schematic designs for this board were adapted from previous designs of KRUPS projects here at the University of Kentucky. Battery charging, improved activation circuitry, a newer IMU, and wireless debug capability are the main additions to previous designs. Newer thermocouple conversion ICs were also added to replace the EOL product that was in previous designs. Activation subsytems and criteria are also outlined.

The following sections outline the electrical connections for control of the board w.r.t. the Teensy 3.5 microcontroller, as well as several relevant subsystem specifications and links to datasheets. Charging and switch wiring for activation are also explained. Schematics are in Appendix A, along with Teensy 3.5 reference card images.

1.1 Primary Activation

Primary activation is triggered by a pin pull out the KREPE enclosure performed by astronauts. Once the pin is pulled, the flight computer is powered on and in standy mode, consuming a minimal amount of power. No radios are powered on in standy mode to ensure no interference with ISS activities.

The `POWER_SW` header must closed for protected battery or USB voltage to be applied to the Teensy's VIN pin, powering on the system. The location of these connection points can be seen in Fig. 2 labelled on the silk screen in the left middle of the PCB. A rendering of the bottom of the board is shown in Fig. 3.

An end-to-end schematic showing battery protection and device activation is shown in Fig. 1. 20 AWG non silicone insulated wire of length less than 5 inches is used to connect the batteries to the first power function (battery protection circuitry).

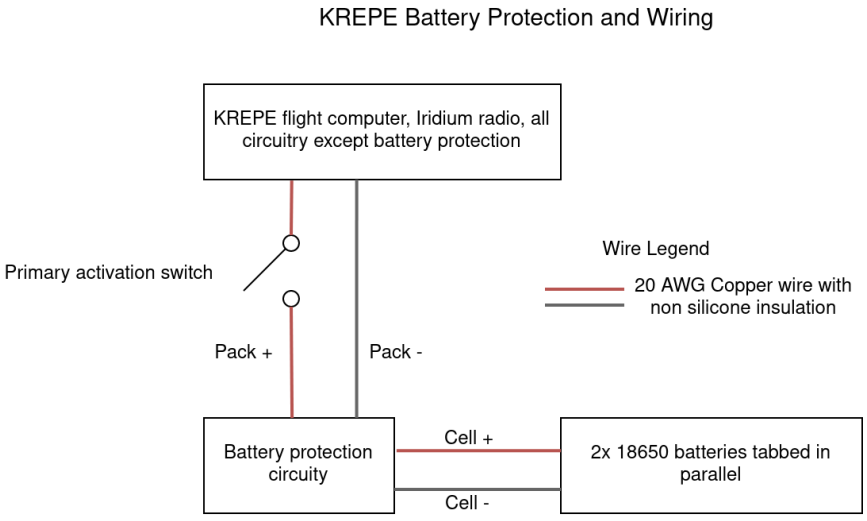


Figure 1: Activation and battery protection schematic overview.

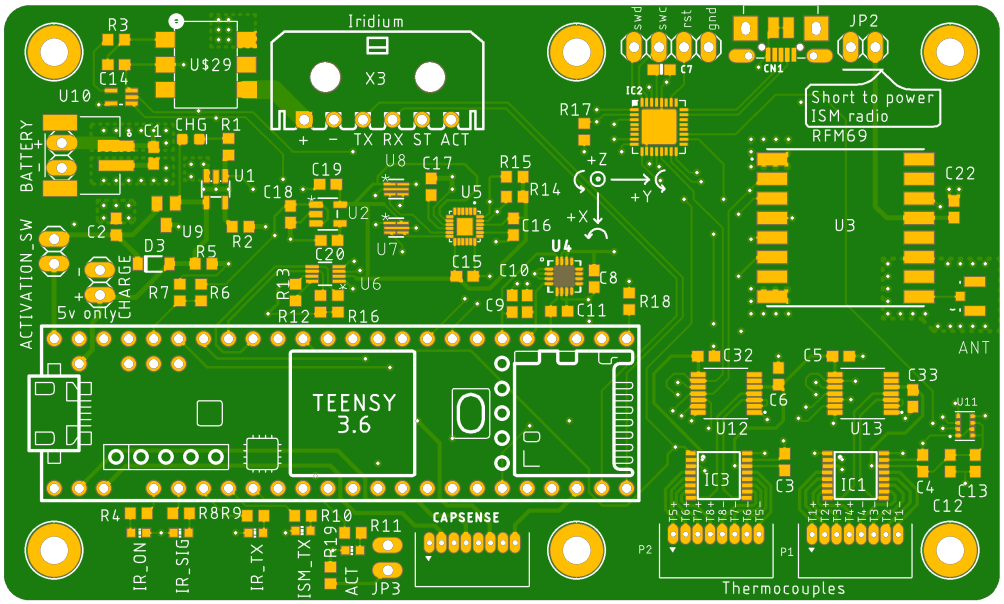


Figure 2: Rendering of the top of the KREPE control board, V1.1.

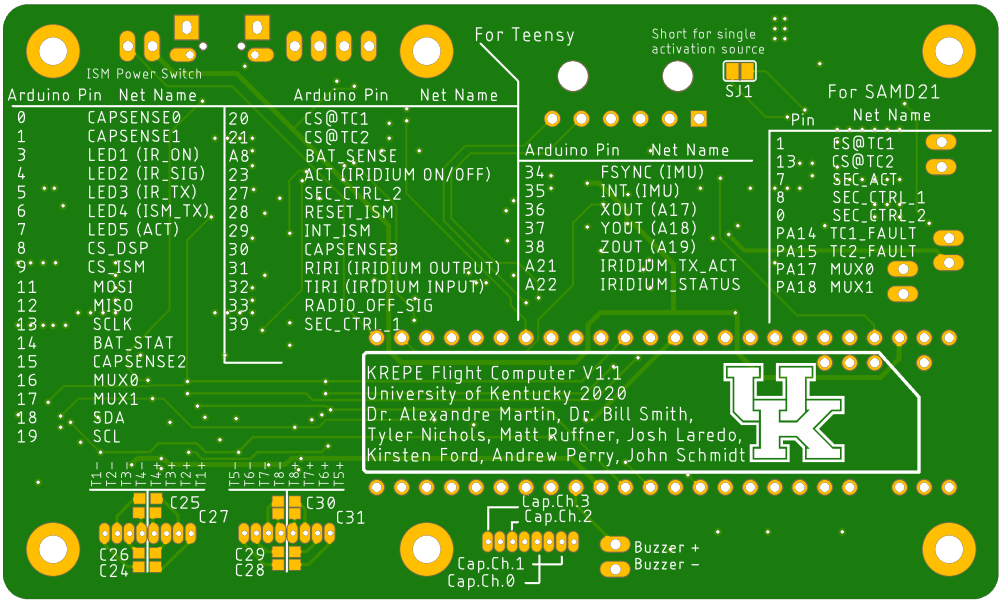


Figure 3: Rendering of the bottom of the KREPE control board, V1.1.

The ISM_SW header is meant to enable and disable the RFM69 debug radio. The center 3.3V pin of this header is connected to the normally closed labeled pin, a GPIO pin is pulled high (see Fig. 3). When the normally closed pin is connected to the center pin, the RFM69 is enabled. This way, debug communication can be used while testing in a way that also ensures it will be off when on a live mission. This radio is only used for ground testing communication purposes, and once handed over for final integration, will never be enabled or able to receive power. These[1] are the switches used for the pin pull activation.

1.2 Secondary Activation

Once primary activation is complete and the flight computer is in standby mode, sensors are polled to check for conditions necessary for secondary activation. Secondary activation is software based and only engaged once the KREPE probe has separated from its protective metal enclosure. No radio transmissions are attempted before secondary activation.

Thermocouples and the capacitive sensing subsystem are polled to check for conditions sufficient for secondary activation. A heating of the metal KREPE enclosure is necessary to melt the plastic bolts that hold it together. This ambient temperature increase of the probe is the primary criteria for secondary activation. The presence of this metal enclosure is also detected by capacitive sensors on the KREPE probe. Once the thermal and capacitive sensing subsystems have detected the separation of the metal enclosure, the Iridium radio is powered on and packet transmission begins.

An activation redundancy processor (ARP) was added in Rev. 1.1 of the flight computer, if the flight computer needs to be tested without the ARP, there is a solder jumper on the bottom of the PCB (SJ1)

[1] https://www.digikey.com/product-detail/en/omron-electronics-inc-emc-div/D2SW-3L1H/Z12268-ND/1811989

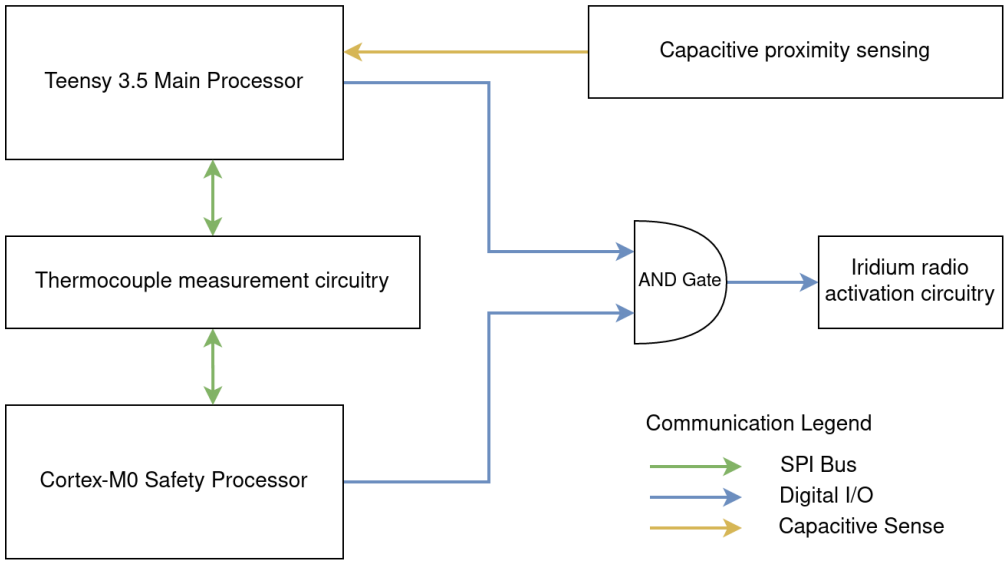


Figure 4: Secondary activation rendundancy provided by the safety processor.

that, when shorted together, bypasses the AND gate that controls Iridium radio activation, reducing the system to single activation.

2 Activation Redundancy Processor

Revision 1.1 of the KREPE flight computer adds a SAMD21 Cortex-M0 safety processor to also monitor the status and reading of all thermocouples onboard the KREPE capsule. Section 2 discusses this more. An overview of the secondary activation redundancy this safety processor provides is shown in Fig. 4.

With the ARP in place, there are two separate CBCS in place in the KREPE capsule, meaning that an erroneous act from one CBCS is not sufficient to activate the iridium radio.

2.1 Bootloading the ARP

The SAMD21 must be flashed with the UF2 bootloader https://learn.adafruit.com/programming-microcontrollers-us

show picture of header for programming, reset pin, swclk, swdio pins.

2.2 Programming the ARP

The microUSB port next to the ARP (CN1) is used to upload program flash using the Arduino IDE. This is only possible once the bootloader has been flashed with OpenOCD.

2.3 ARP Hardware Connections

Show which pins connect into the SPI bus and which goes to the AND gate for iridium activation.

3 Subsystems

3.1 Status and Error Indicators

Teensy Pin	Net Name	Teensy Configuration
3	LED1 - IRIDIUM ON	OUTPUT
4	LED2 - IRIDIUM SIGNAL OK	OUTPUT
5	LED3 - IRIDIUM RADIO TRANSMITTING	OUTPUT
6	LED4 - ISM RADIO TRANSMITTING	OUTPUT
7	LED5 - GENERAL ACTIVITY	OUTPUT

Table 1: Debug LED Connections.

3.2 Serial Interface Signals

Teensy Pin	Net Name	Description
13	SCLK	SPI Clock
12	MISO	Master In Subject Out
11	MOSI	Master Out Subject In
20	CS@TC1	MAX31856 chip select, active low
21	CS@TC2	MAX31856 chip select, active low
9	CS_ISM	RFM69 chip select, active low
32	TIRI	Iridium TX UART
31	RIRI	Iridium RX UART
19	SCL	I^2C bus clock
18	SDA	I^2C bus data

Table 2: Pins used with SPI, I^2C, and UART interfaces.

3.3 RFM69 Radio

Note that this radio is not supplied with power unless the NO to C connection is made on the ISM_SW header (see Fig. 2). Maximum output power according to the radio datasheet (https://cdn.sparkfun.com/datasheets/Wireless/General/RFM69HCW-V1.1.pdf) is 100mW.

Teensy Pin	Net Name	Description	Teensy Configuration
28	RESET_ISM	Pull low to enable RFM69	OUTPUT
29	INT_ISM	GPIO0 interrupt from RFM69	INPUT
33	RADIO_OFF_SIG	Pulled high when the RFM69 is disabled	INPUT

Table 3: Radio module interface signals.

The datasheet for this antenna can be found at https://cdn.taoglas.com/datasheets/FXP290.07.0100A.pdf.

3.4 Iriduim Radio

We are using the A3LA-RS type modem seen on the NAL Research site (http://www.nalresearch.com/IridiumHardware.html). The RF specifications, taken from the module's datasheet are shown in Fig. 5.

Operating Frequency:	1616 to 1626.5 MHz
Duplexing Method:	TDD
Multiplexing Method:	TDMA/FDMA
Link Margin:	12 dB average
Average Power during a Transmit Slot (Max):	7W
Average Power during a Frame (Typical):	0.6W
Receiver Sensitivity at 50Ω (Typical):	−118 dBm

Figure 5: RF specifications of the AL3A-RS Iridium modem.

3.4.1 Radio Power Control

This is the partial activation source for the Iridium radio. Rev. 1.1 the flight computer features a secondary safety processor to continually monitor the thermocouple measurement circuits in parallel with the Teensy. An AND gate controls the power to the solid state relay controlling power to the Iridium radio. THe secondary processor is connected to the other input of this AND gate to make sure that erroneous action on behalf the teensy (or secondary safety processor) is not able to solely activate the radio.

The following table shows the pin from the Teensy that activates the solid state relay controlling power to the Iridium radio.

Teensy Pin	Net Name	Description	Teensy Configuration
23	PRI_ACT	Primary Iridium activation, active high	`OUTPUT`

Table 4: Pin controlling power to the iridium satellite radio.

3.5 Thermocouple Measurement Interface

Note: this board features an update thermocouple interface IC than the previous boards. Among other enhancements it allows for broader temperature range reading and improved precision.

Teensy Pin	Net Name	Description	Teensy Configuration
16	MUX0	MUX select pin 0	`OUTPUT`
17	MUX1	MUX select pin 1	`OUTPUT`
25	TC1_FAULT	U13 fault (active low)	`INPUT`
26	TC2_FAULT	U12 fault (active low)	`INPUT`

Table 5: Analog mux selection and thermocouple fault status pins.

3.5.1 Thermocouple Connections

The 8 thermocouple connections are done with 2 analog multiplexers IC1 and IC3 (MUX1 and MUX2). The MUX select pins go to both of these chips to select a certain channel. The table of MUX(0/1) select values versus two selected thermocouples are shown in Table 6.

MUX0	MUX1	TC number
0	0	1, 5
0	1	2, 6
1	0	3, 7
1	1	4, 8

Table 6: Truth table for multiplexer select pins and their relation to the pairs of thermocouples that are selected.

Pin connections on headers P1 and P2 show the connections for TC 1-8 lead wire pairs. Figure 6 shows the pinout on the silkscreen.

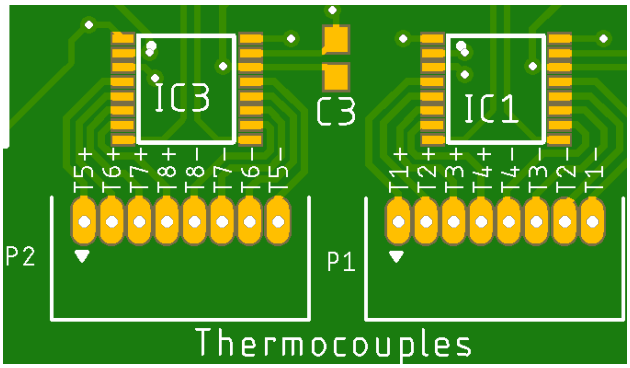


Figure 6: Thermocouple connection wiring with resepect to the analog mux chips IC3 (MUX2) and IC1 (MUX1).

3.6 Barometric Pressure Sensor Connections

Version 1.1 of the flight computer features an Infineon DSP310 barometric pressure and temperature sensor connected to the SPI bus with active low chip select on pin 8 of the teensy.

3.7 Motion Sensor Connections

Teensy Pin	Net Name	Description	Teensy Configuration
36 A17	XOUT	Analog out from accel (x axis)	`INPUT`
37 A18	YOUT	Analog out from accel (y axis)	`INPUT`
38 A19	ZOUT	Analog out from accel (z axis)	`INPUT`
35	INT	Interrupt from ICM-20948	`INPUT`
34	FSYNC	Synchronization signal to ICM-20948	`OUTPUT`

Table 7: Pins connecting to the ADXL377 and ICM-20948.

3.8 Charging and Power

Charge current is limited to to 450 mA. Charge power can be delivered via Teensy USB or the `CHARGE` header. Charging input voltage is expected to be 5 volts.

For battery protection, the adafruit batteries we use (https://www.adafruit.com/product/354) have built in protection circuitry. Charge management is handled by an MCP73831 IC (https://www.microchip.com/wwwproducts/en/MCP73831), with status connections to the Teensy as shown in Table 8. Schematics and electrical connections are shown in Fig. 12 in Appendix A.

3.8.1 Battery Status Interface

Teensy Pin	Net Name	Description	Teensy Configuration
14	BAT_STAT	LiPo charge state	`OUTPUT`
22 A8	BAT_SENSE	Halved battery voltage for monitoring	`INPUT`

Table 8: Pins to monitor battery voltage and charging status.

3.8.2 Battery Protection

Protection circuitry is implemented on the flight board to support 2P1S LiPo packs for system power. We are using a TI BQ2970 Voltage and Current Protection IC (http://www.ti.com/lit/ds/symlink/bq2970.pdf). Protection circuitry as implemented on the KREPE flight computer PCB is shown in Fig. 7. Cell+ and Cell- attach to the battery pack and Pack+/Pack- face system power. This protection circuitry is upstream of the primary activation switch.

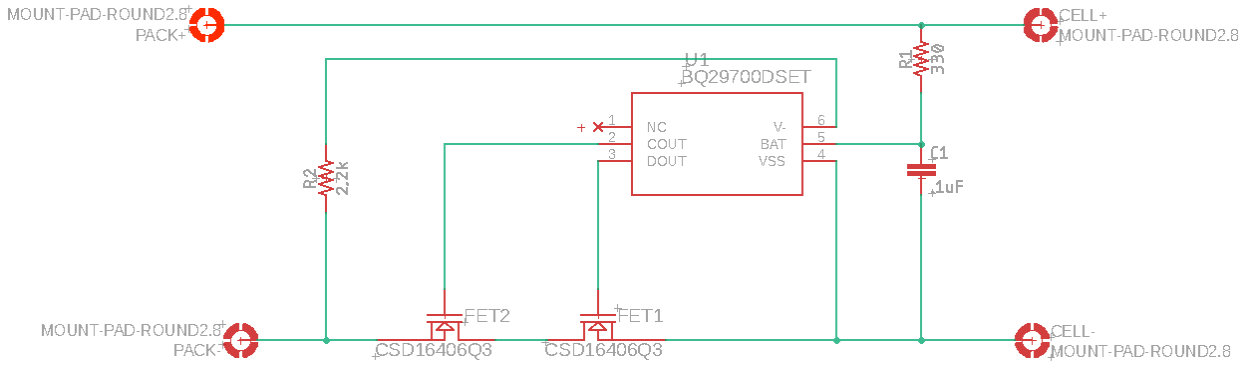


Figure 7: Battery protection circuitry.

Renderings of the bottom and top of the battery protection PCB can be seen in Figs 8 and 9.

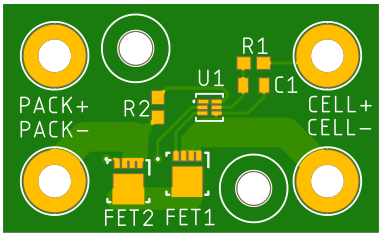


Figure 8: Rendering of the top of the battery protection PCB.

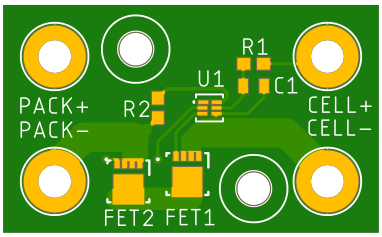


Figure 9: Rendering of the bottom of the battery protection PCB.

A Schematics

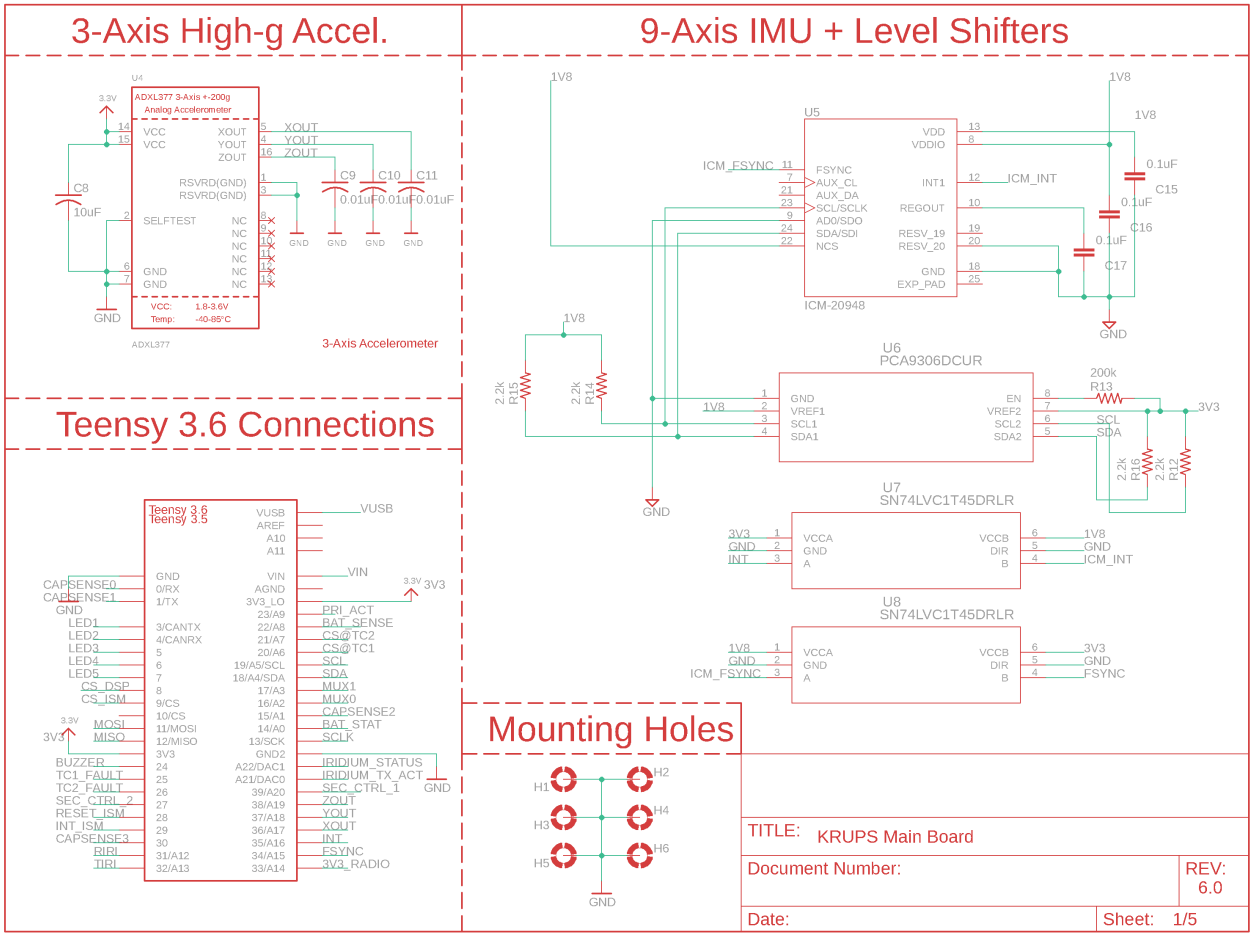


Figure 10: Page one of schematics.

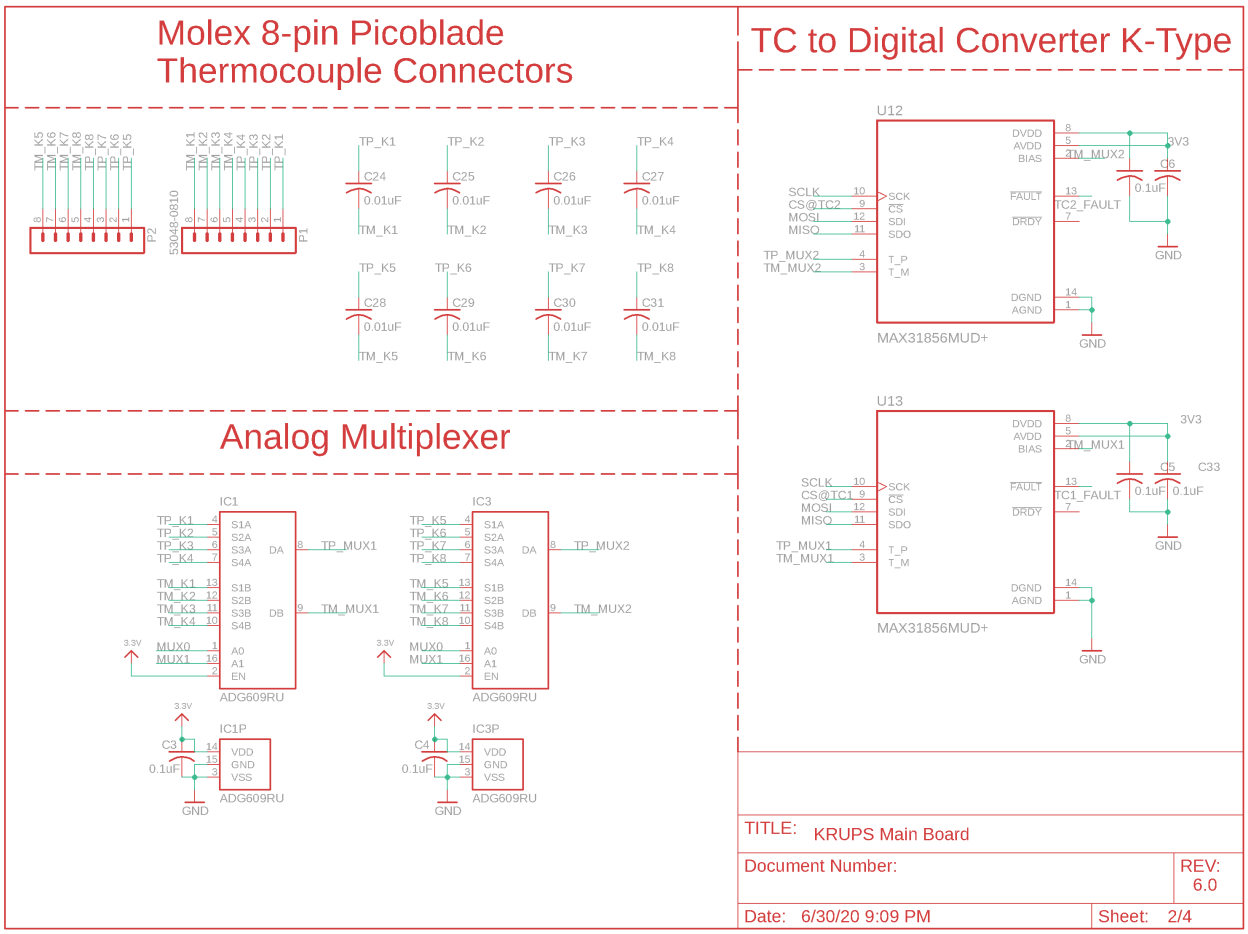


Figure 11: Page two of schematics.

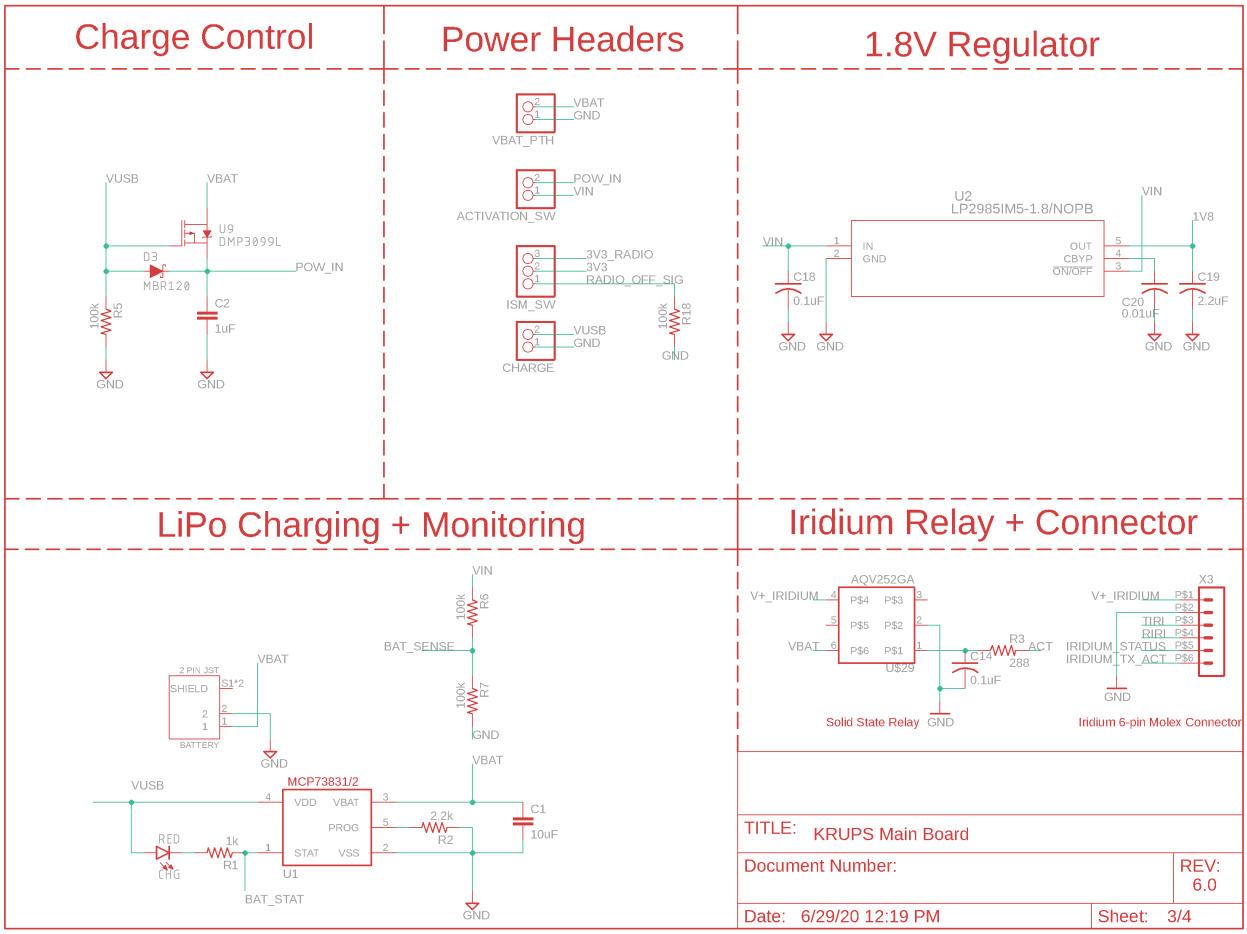


Figure 12: Page three of schematics.

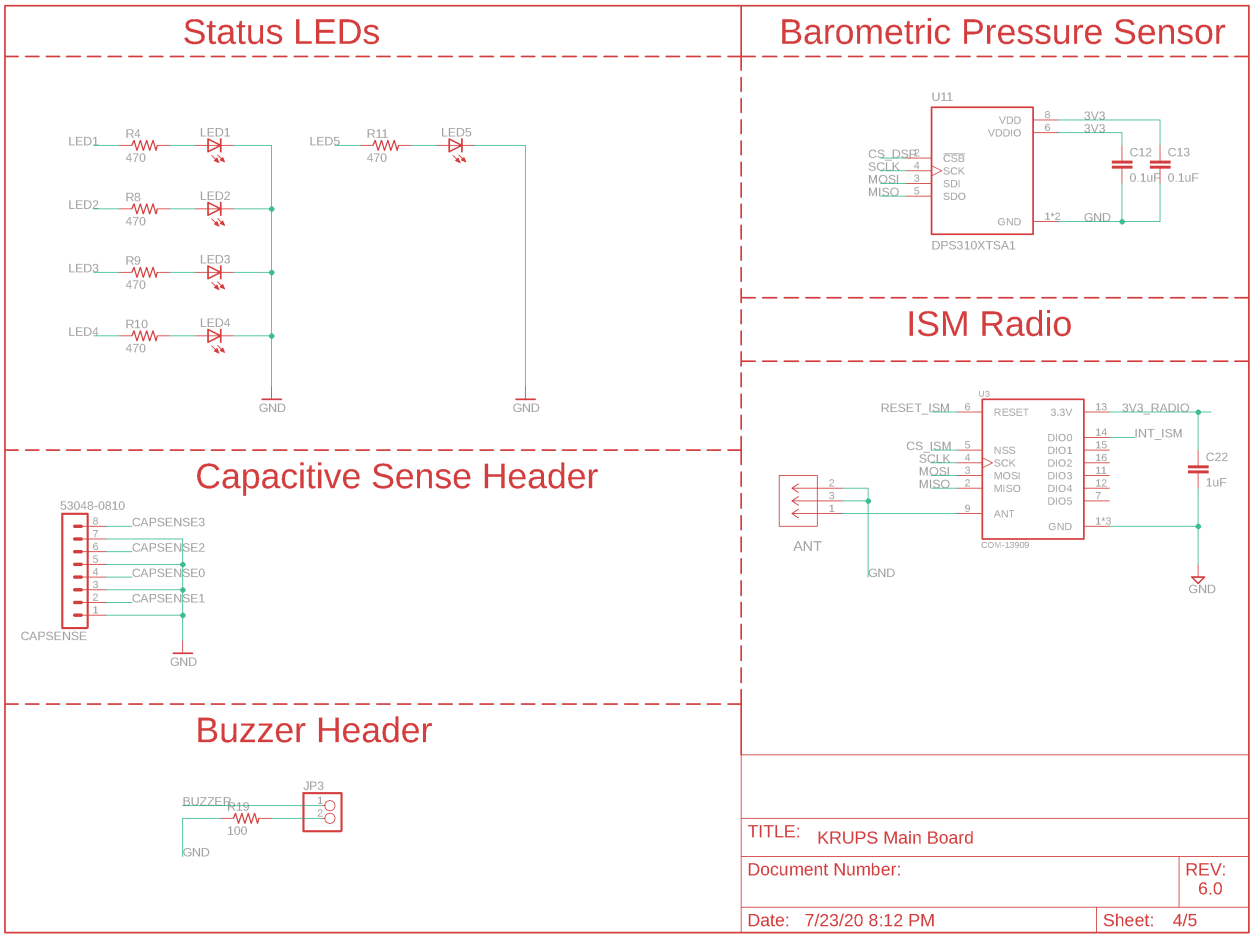


Figure 13: Page four of schematics.

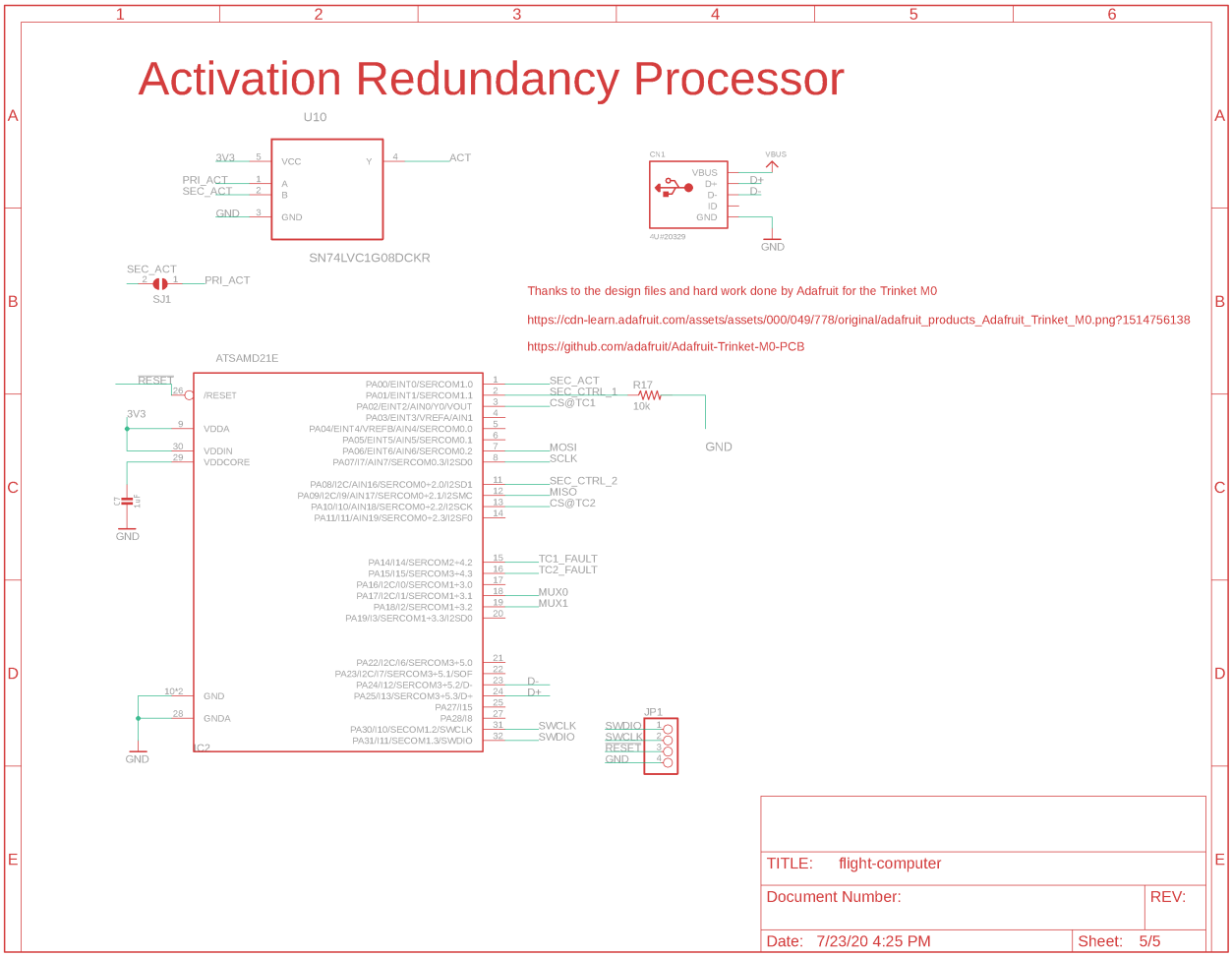


Figure 14: Page five of schematics.

B Teensy 3.5 Reference

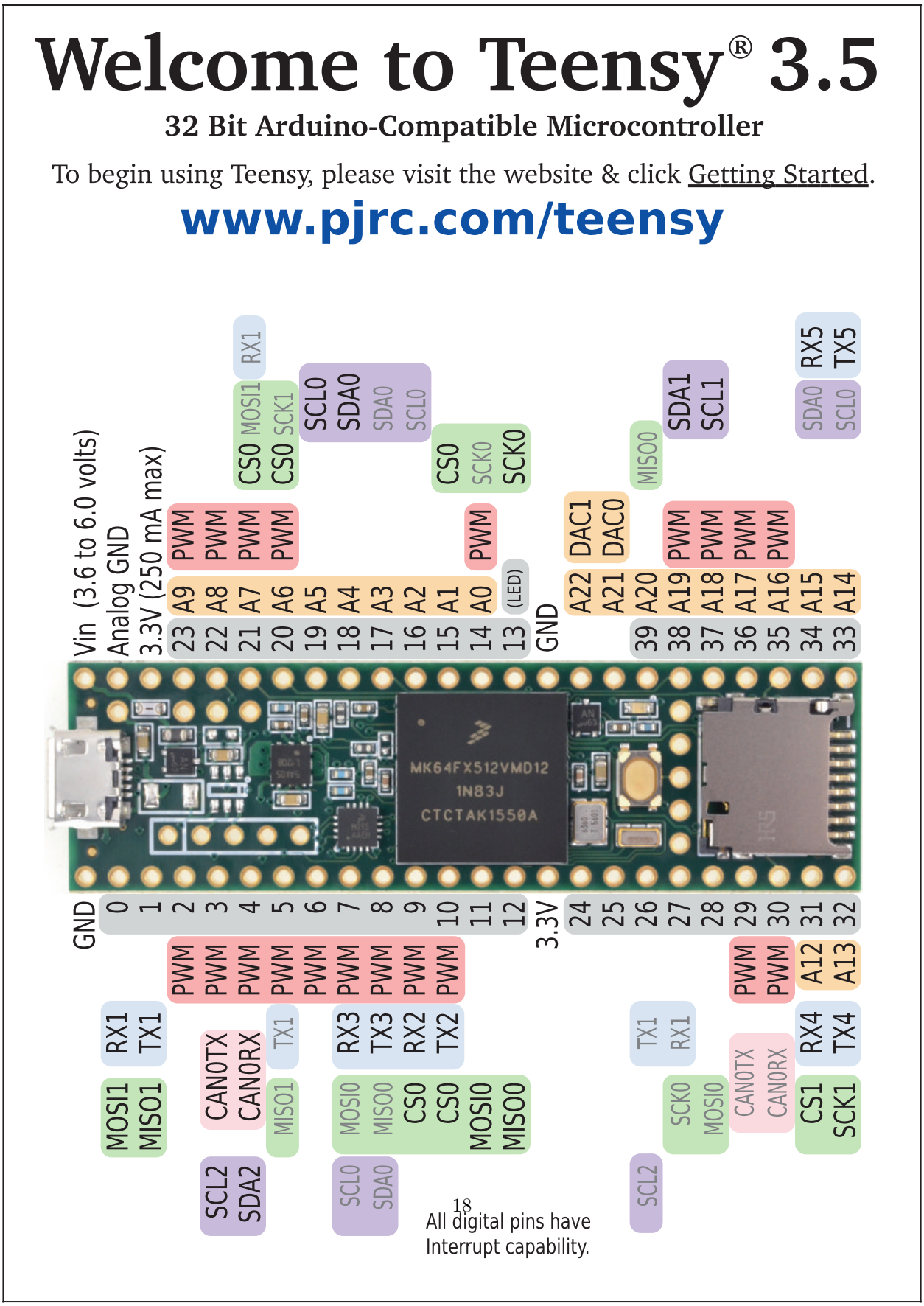
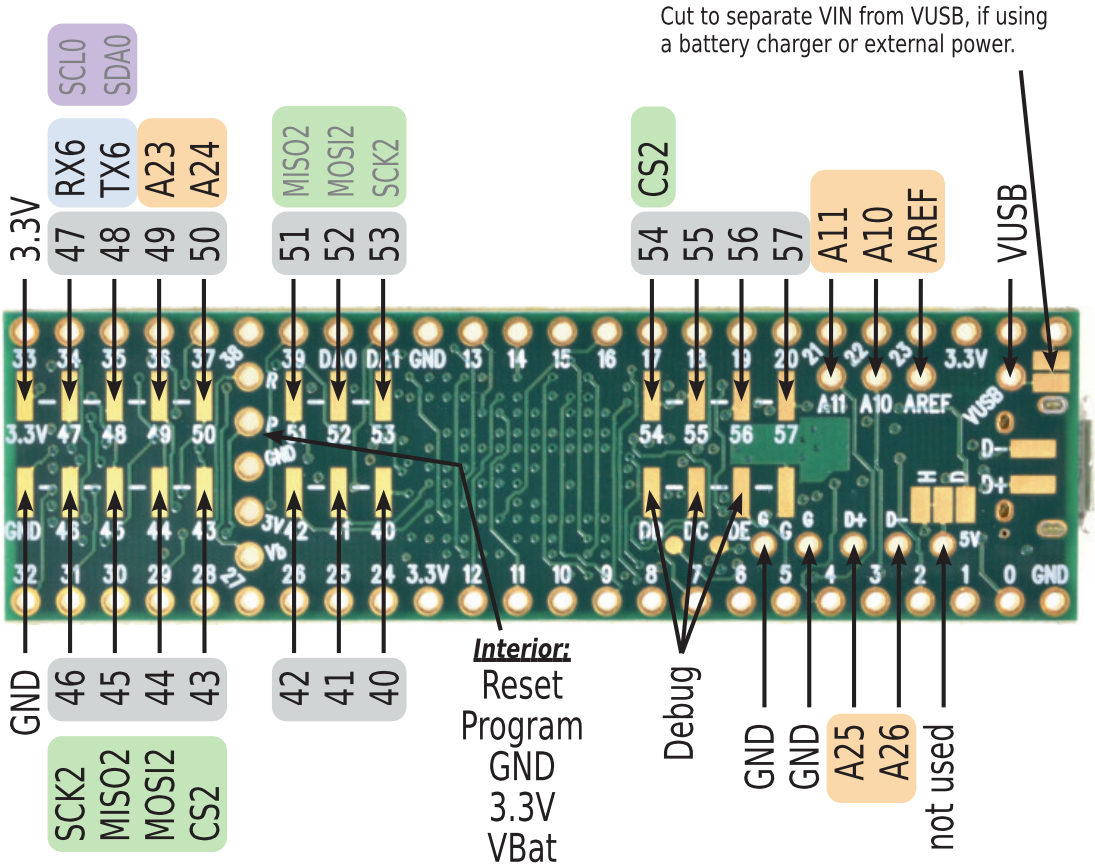


Figure 15: Teensy 3.5 Front

Teensy® 3.5 Back Side

Additional pins and features available on the back side



Teensy 3.5 pins with digital I/O are 5 volt tolerant. Other pins are **not** 5V tolerant. Do not apply more than 3.3V to A10, A11, A21, A22, A25, A26, AREF, Program or Reset.

For solutions to the most common issues and technical support, please visit:

www.pjrc.com/help

Teensy 3.5 System Requirements:
PC computer with Windows 7, 8, 10 or later
or Ubuntu Linux 12.04 or later
or Macintosh OS-X 10.7 or later
USB Micro-B Cable

Figure 16: Teensy 3.5 Back

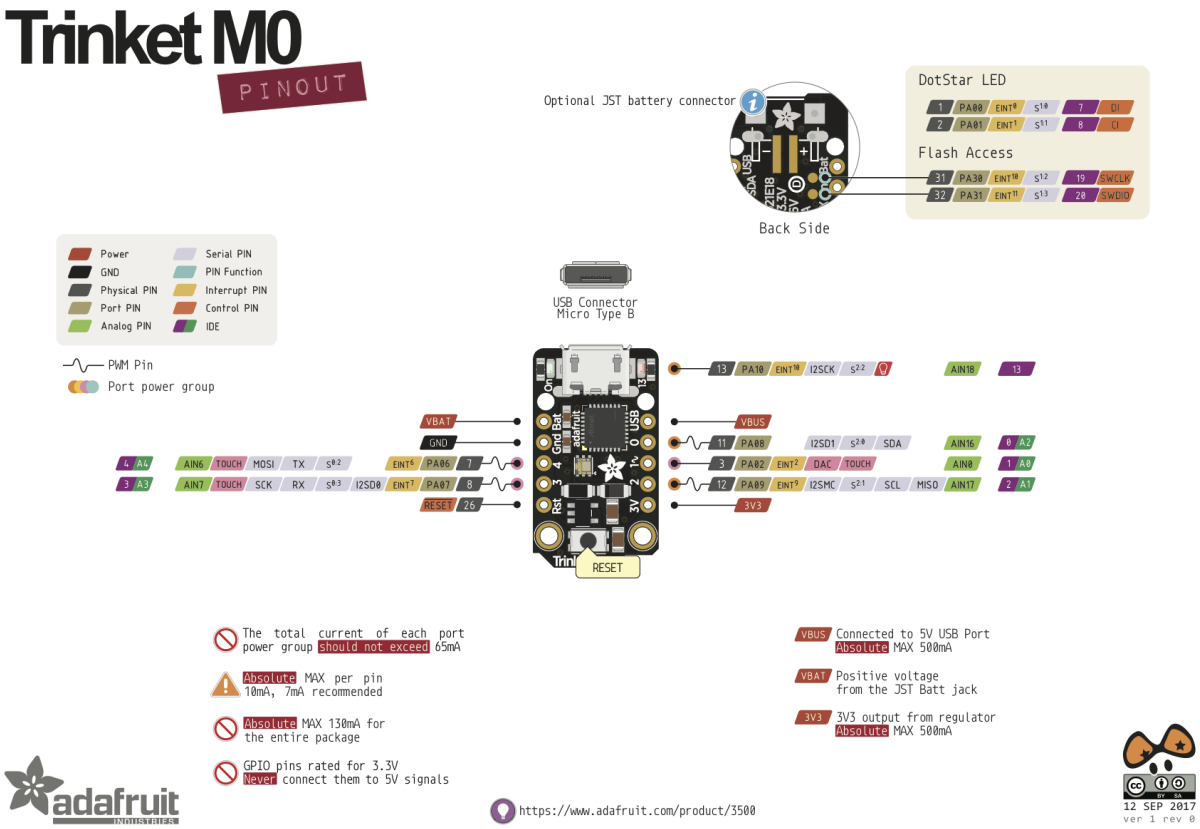


Figure 17: Trinket-M0 Arduino Reference (for ARP chip).

C Partslist

Partlist

Exported from flight-computer.sch at 7/23/20 8:11 PM

EAGLE Version 9.6.2 Copyright (c) 1988-2020 Autodesk, Inc.

Assembly variant:

Part	Value	Device	Package	Library	Sheet
ACTIVATION_SW		PINHD-1X2	1X02	pinhead	3
ANT	U.FL-R-SMT-1(10)	U.FL-R-SMT-1(10)	CONN_R-SMT-1(10)	ufl	4
BATTERY	2 PIN JST	S2B-PH-SM4-TB(LF)(SN)	JST_S2B-PH-SM4-TB(LF)(SN)	S2B-PH-SM4-TB_LF__SN_	3
C1	10uF	C-EUC0603	C0603	rcl	3
C2	1uF	C-EUC0603	C0603	rcl	3
C3	0.1uF	C-USC0603	C0603	adafruit	2
C4	0.1uF	C-USC0603	C0603	adafruit	2
C5	0.1uF	C-USC0603	C0603	adafruit	2
C6	0.1uF	C-USC0603	C0603	adafruit	2
C7	1uF	CAP_CERAMIC0603_NO	0603-NO	microbuilder	5
C8	10uF	C-USC0603	C0603	adafruit	1
C9	0.01uF	C-USC0603	C0603	adafruit	1
C10	0.01uF	C-USC0603	C0603	adafruit	1
C11	0.01uF	C-USC0603	C0603	adafruit	1
C12	0.1uF	C-EUC0603	C0603	rcl	4
C13	0.1uF	C-EUC0603	C0603	rcl	4
C14	0.1uF	C-USC0603	C0603	adafruit	3
C15	0.1uF	C-EUC0603	C0603	rcl	1
C16	0.1uF	C-EUC0603	C0603	rcl	1
C17	0.1uF	C-EUC0603	C0603	rcl	1
C18	0.1uF	C-USC0603	C0603	rcl	3
C19	2.2uF	C-USC0603	C0603	rcl	3
C20	0.01uF	C-USC0603	C0603	rcl	3
C22	1uF	C-EUC0603	C0603	rcl	4
C24	0.01uF	C-USC0603	C0603	adafruit	2
C25	0.01uF	C-USC0603	C0603	adafruit	2

C26	0.01uF	C-USC0603	C0603	adafruit	2
C27	0.01uF	C-USC0603	C0603	adafruit	2
C28	0.01uF	C-USC0603	C0603	adafruit	2
C29	0.01uF	C-USC0603	C0603	adafruit	2
C30	0.01uF	C-USC0603	C0603	adafruit	2
C31	0.01uF	C-USC0603	C0603	adafruit	2
C32	0.1uF	C-USC0603	C0603	adafruit	2
C33	0.1uF	C-USC0603	C0603	adafruit	2
CAPSENSE	53048-0810	53048-0810	53048-0810	con-molex-picoblade	4
CHARGE		PINHD-1X2	1X02	pinhead	3
CHG	RED	LED-RED0603	LED-0603	SparkFun-LED	3
CN1	4U#20329	USB_MICRO_20329_V2	4UCONN_20329_V2	microbuilder	5
D3	MBR120	MBR120	SOD123FL	gsynth	3
H1	MOUNT-PAD-ROUND2.8	MOUNT-PAD-ROUND2.8	2,8-PAD	holes	1
H2	MOUNT-PAD-ROUND2.8	MOUNT-PAD-ROUND2.8	2,8-PAD	holes	1
H3	MOUNT-PAD-ROUND2.8	MOUNT-PAD-ROUND2.8	2,8-PAD	holes	1
H4	MOUNT-PAD-ROUND2.8	MOUNT-PAD-ROUND2.8	2,8-PAD	holes	1
H5	MOUNT-PAD-ROUND2.8	MOUNT-PAD-ROUND2.8	2,8-PAD	holes	1
H6	MOUNT-PAD-ROUND2.8	MOUNT-PAD-ROUND2.8	2,8-PAD	holes	1
IC1	ADG609RU	ADG609RU	TSSOP16	analog-devices	2
IC2	ATSAMD21E	ATSAMD21E	QFN32_5MM	microbuilder	5
IC3	ADG609RU	ADG609RU	TSSOP16	analog-devices	2
JP1		PINHD-1X4	1X04	pinhead	5
JP2		PINHD-1X2	1X02	pinhead	3
JP3		PINHD-1X2	1X02	pinhead	4
LED1		LEDCHIP-LED0603	CHIP-LED0603	adafruit	4
LED2		LEDCHIP-LED0603	CHIP-LED0603	adafruit	4
LED3		LEDCHIP-LED0603	CHIP-LED0603	adafruit	4
LED4		LEDCHIP-LED0603	CHIP-LED0603	adafruit	4
LED5		LEDCHIP-LED0603	CHIP-LED0603	adafruit	4
P1	53048-0810	53048-0810	53048-0810	con-molex-picoblade	2
P2	53048-0810	53048-0810	53048-0810	con-molex-picoblade	2
R1	1k	R-US_R0603	R0603	rcl	3
R2	2.2k	R-US_R0603	R0603	rcl	3
R3	288	R-US_R0603	R0603	adafruit	3
R4	470	R-US_R0603	R0603	rcl	4
R5	100k	R-US_R0603	R0603	rcl	3
R6	100k	R-US_R0603	R0603	rcl	3
R7	100k	R-US_R0603	R0603	rcl	3
R8	470	R-US_R0603	R0603	rcl	4
R9	470	R-US_R0603	R0603	rcl	4
R10	470	R-US_R0603	R0603	rcl	4
R11	470	R-US_R0603	R0603	rcl	4
R12	2.2k	R-US_R0603	R0603	rcl	1
R13	200k	R-US_R0603	R0603	rcl	1
R14	2.2k	R-US_R0603	R0603	rcl	1
R15	2.2k	R-US_R0603	R0603	rcl	1
R16	2.2k	R-US_R0603	R0603	rcl	1
R17	10k	R-US_R0603	R0603	rcl	5
R18	100k	R-US_R0603	R0603	rcl	3
R19	100	R-US_R0603	R0603	rcl	4
SJ1		SJ	SJ	jumper	5
U$1	TEENSY_3.5/3.6_BASIC	TEENSY_3.5/3.6_BASIC	TEENSY_3.5/3.6_BASIC	Teensy356	1
U$29	AQV252GA	AQV252GA	DIP6	TI_radio	3
U1	MCP73831/OT	MCP73831/OT	SOT23-5L	adafruit	3
U2	LP2985IM5-1.8/NOPB	LP2985IM5-1.8/NOPB	MF05A	gsynth	3
U3	COM-13909	COM-13909	MOD_COM-13909	COM-13909	4
U4	ADXL377	ACCEL_ADXL377	LFCSP16_LQ	microbuilder	1
U5	ICM-20948	ICM-20948	QFN40P300X300X105-25N	ICM-20948	1
U6	PCA9306DCUR	PCA9306DCUR	DCU8	gsynth	1
U7	SN74LVC1T45DRLR	SN74LVC1T45DRLR	DRL6	gsynth	1
U8	SN74LVC1T45DRLR	SN74LVC1T45DRLR	DRL6	gsynth	1
U9	DMP3099L	DMP3099L	SOT23	gsynth	3
U10	SN74LVC1G08DCKR	SN74LVC1G08DCKR	SOT65P210X110-5N	SN74LVC1G08DCKR	5
U11	DPS310XTSA1	DPS310XTSA1	XDCR_DPS310XTSA1	DPS310XTSA1	4
U12	MAX31856MUD+	MAX31856MUD+	SOP65P640X110-14N	MAX31856	2
U13	MAX31856MUD+	MAX31856MUD+	SOP65P640X110-14N	MAX31856	2
VBAT_PTH		PINHD-1X2	1X02	pinhead	3
X3		HEADER_POS6_43650-0600	43650-0600	con-molex-micro-fit-3_0	3

D Arduino Pin Mapping

Arduino Pin	Net
Teensy 3.5	
0	CAPSENSE0
1	CAPSENSE1
3	LED_IRIDIUM_ON
4	LED_IRIDIUM_SIGNAL_OK
5	LED_IRIDIUM_TRANSMITTING
6	LED_ISM_TRANSMITTING
7	LED_ACTIVITY
8	CS_DSP
9	CS_ISM
11	MOSI
12	MISO
13	SCLK
14	BAT_STAT
15	CAPSENSE2
16	MUX0
17	MUX1
18	SDA
19	SCL
20	CS@TC1
21	CS@TC2
A8	BAT_SENSE (A8)
23	PRI_ACT (IRIDIUM ON/OFF)
24	BUZZER
25	TC1_FAULT (ACTIVE LOW)
26	TC2_FAULT (ACTIVE LOW)
27	SEC_CTRL_2
28	RESET_ISM
28	INT_ISM
30	CAPSENSE3
31	RIRI (IRIDIUM OUTPUT)
32	TIRI (IRIDIUM INPUT)
33	RADIO_OFF_SIG
34	FSYNC (IMU)
35	INT (IMU)
36	XOUT (A17)
37	YOUT (A18)
38	ZOUT (A19)
39	SEC_CTRL_1
A21	IRIDIUM_TX_ACT
A22	IRIDIUM_STATUS
Safety Processor (ATSAMD21E16B)	
1	CS@TC1
13	CS@TC2
7	SEC_ACT
8	SEC_CTRL_1

0	SEC_CTRL_2
PA14	TC1_FAULT
PA15	TC2_FAULT
PA17	MUX0
PA18	MUX1

E General Requirements Compliance

Upon both power up (after primary activation via pull-tab) and detection of a termination or off-nominal power condition, the Teensy processor will enter a safe reset state, where all GPIO pins controlling critical system functions are set to a high impedance value. This high impedance state, in addition to the flight computer PCB, ensure requirements in the *General Requirements for the Computer-Based Control System Safety Requirements for the ISS* are met. Overcurrent and undervoltage protection for battery cells is implemented upstream of the flight computer, limiting the off-nominal power conditions expected (see KREPE Flight Computer Hardware Manual). See Fig. 18 for an overview of the main execution lifecycle and how upon both power up and detection of an abnormal power condition both put the CBCS back into the safe high impedance state. CBCS General Requirements are discussed below.

E.1 Req. 3.1.1.1

Teensy 3.5 controller documentation shows that the controller powers up into the known safe reset state. No outputs occur until the processing state is initiated (see Fig. 18). Verification for this requirement will monitor the state of Teensy output pins with external hardware upon power up to make sure specific signals (i.e. iridium soli state relay activation signal) do not transition unexpectedly.

E.2 Req. 3.1.1.2

KREPE flight computer software on the Teensy 3.5 controller enters a safe state in the event that a termination condition is detected (e.g. low system battery voltage; see Fig. 18). As the KREPE capsule is incapable of receiving any external commands, the detection of a termination command scenario is not applicable. Testing for compliance with this requirement entails monitoring of Flight computer pin state with external hardware while the system experiences low battery voltage to make sure it places critical pins in a high impedance or off state.

E.3 Req. 3.1.1.3

The KREPE battery protection subsystem prevents any overcurrent or undervoltage conditions from reaching the flight computer. The main off-nominal power condition the flight computer may encounter is power failure. The Teensy 3.5 controller features a power management controller that will place it in a safe reset state if a power failure is detected. The KREPE flight computer also monitors system battery voltage to preemtively detect an off-nominal power condition and place the system in the safe state (see Fig. 18).

Testing of all flown cells will be done including full charge/discharging cycling, under-voltage, over-voltage, and over-current testing with the battery protection circuitry in place. Analysis will be performed via debug interface and external hardware to make sure the flight computer returns to the prescribed safe reset state upon recovery from these off-nominal power conditions.

E.4 3.1.1.4

N/A

E.5 3.1.1.5

N/A

E.6 3.1.1.6

The metal enclosure that houses the KREPE module acts as a Faraday cage and mitigates the risk imposed to the CBCS by inadvertent memory modification. Inspection See 3.1.1.4 Suggest putting a transmitter inside and see if we can detect any leakage at MSFC if possible.

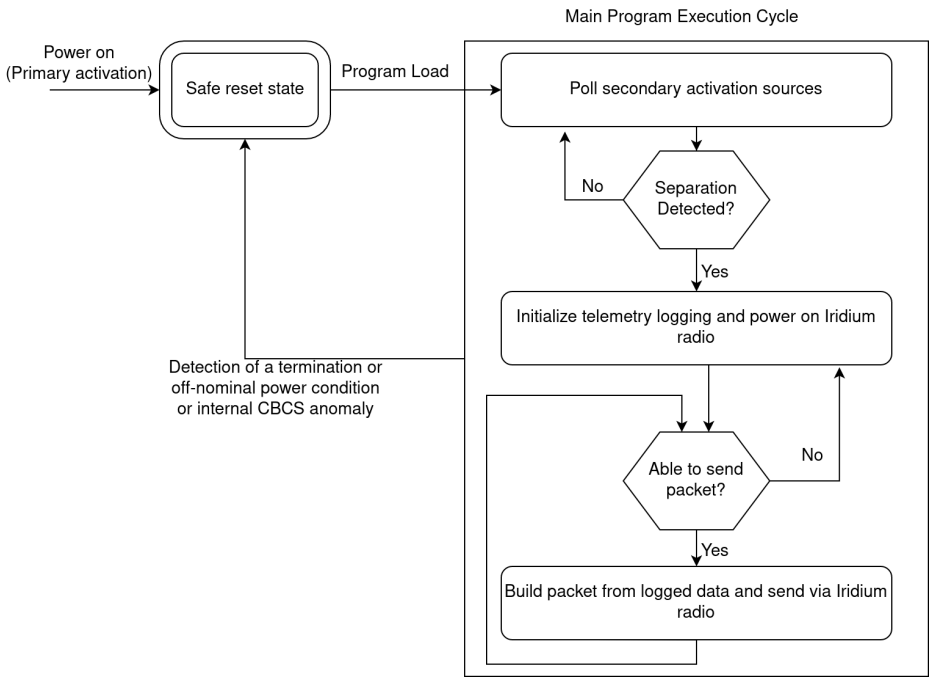


Figure 18: Overview of main execution cycle showing return to safe high-Z reset state upon startup and abnormal power condition.

E.7 3.1.1.7

Upon detection of an anomaly internal to the CBCS there is watchdog timer functionality that will return the CBCS to the known safe initialization state. Verification procedures for this requirement entail physical disruption/disconnection of flight computer hardware to ensure control software can recover.

E.8 3.1.1.8

The external sources in this case would be the ambient temperature of capsule and the presence of the metal enclosure around the KREPE capsule. The capsule uses the Fault flags on the thermocouple to digital converter are used to discern if a valid thermocouple reading has been collected or not. There are also multiple thermocouples whos values are cross checked for consistency in valid readings. Capacitive sensing is also used to detect the presence of the metal enclosure around the KREPE capsule. Both of these inputs are used to discern between valid and invalid input.

Verification for this requirement entails physical disconnection/perturbation of thermocouple and capacitive sensing leads to analyze the induced effect on measurement values and fault conditions.

E.9 3.1.1.9

Inspection Inspection of software code verifies all lines of code are traceable to system or software requirements. Test Coverage analysis of flight code will ensure lack of dead code and all system software acts for a requirement. See Fig. 18 for KREPE software design requirements.

E.10 3.1.1.10

All flight software is developed to first meet system requirements. The system requirements outline what functionality will be required from the software as well as what will be available to serve as sources of infor-

mation. Another consideration that is taken during this phase of software development is the hardware that the software will be running on. As the development cycle gets into writing the software itself, the primary method of configuration management follows a traditional Agile development cycle with team meetings every week to address concerns and defects. The codebase itself is structured and modular in nature so making isolated changes to configure the software to a different hardware or mission spec is controlled and reversible if need be.

Version control is also used to maintain both productivity and preserve proper versioning of the codebase once large sets of requirements are successfully encompassed and have passed testing. For testing and building release versions of the software, all merge requests must be approved by at least 1 other engineer to preserve main version branch functionality and which requirements each version completes will be documented in the repository itself. Verification and validation processes happen under formal testing conditions when hardware is readily available to run simulated mission conditions and ensure the checklist of requirements is fulfilled.

E.11 3.1.1.11

N/A - the single board design of the KREPE module is designed such that complicated transmission and reception lines between devices, e.g. 1553 busses, are not necessary.

E.12 3.1.1.12

N/A - KREPE has no requirement for audio communication. KREPE has no uplink capability, therefore unauthorized third party control is not possible for KREPE.

E.13 3.1.1.13

N/A - KREPE does not receive any commands.