

Designing an Arduino-based Rocket Flight Computer for Embedded Systems Education

Joseph T. Telaak

University of South Carolina, Columbia, SC, 29201, US

Dr. Wout De Backer

Associate Professor, Ronald E. McNAIR Center for Aerospace Innovation and Research, University of South Carolina, Columbia, SC, 29201, US

As amateur model rockets get more advanced, flight computers are included. They offer many benefits and enable users to include more advanced features like data logging, location tracking, thrust vector control (TVC), delayed ignition charge activation, and active component actuation. Of-the-shelf flight computers can be quite expensive, but may offer few customization options. This may lead to budget challenges for amateur rocketry projects. Working directly with complex hardware and low-level software can be intimidating for most, so development boards have been created to make amateur projects more accessible. Therefore, there is a desire to develop a low-cost flight computer suitable for all budgets, which is where microcontroller platforms like Arduino excel. The Arduino platform has many community resources and easy-to-use software, like the Arduino IDE. An inexpensive, full-featured, and modifiable Arduino-based rocket flight computer was designed based on the Arduino Mega microcontroller. This computer is compatible with the Arduino IDE as well as the standard Arduino Hardware Abstraction Layer (HAL). Common sensors like an accelerometer, barometer, and altimeter can provide the user with many different options for flight data logging and active control. On-board memory was also included to store flight data. In addition, the computer can drive high-power components like servos and ignition charges. To make software development more accessible for beginners, a comprehensive software package simplifies access to many of the sensors, communication, and external components, and a simple programming interface can be used to extract the data. Examples of certain control algorithms, like TVC, were developed to demonstrate their capabilities to users.

I. Introduction

When designing an amateur rocket, a flight computer is necessary to collect sensor data and operate active components of the rocket. As an example, if you want to actively deploy a parachute at a certain altitude, you need to be able to read the pressure data to determine altitude, process that data to check for the correct activation altitude, and then activate the charge. There are many commercial options available for users to choose from, but these options can be very expensive. These computers also offer limited options for both the hardware and software customization. Some of these computers can cost over \$300 (See Table 6 for a comparison). These prices are outside of some budgets, so many users attempt to develop their own flight computers instead. Designing your own flight computer offers many opportunities to educate users in circuit design, EDA software, embedded systems programming, and algorithmic design. Getting started in the process of building your own computer can be intimidating, there are many factors to consider and many possibilities for errors that could damage components or leave you with a non-working project. This leaves an opening for a flight computer that is designed to have similar features as commercial flight computers while being open-source, extensible, and user-friendly. The design files will also be available if a user wants to build it themselves. The following sections describe the rationale behind the design choices, the implementation of the components, the software drivers, compares this computer to existing commercial computers.

During the design phase of the project, multiple existing flight computers were used as a reference to select the most desirable features. Computers like the BPS.space Signal[?] and the TeleMetrum[?] are examples of common computers with multiple useful features beyond the basic parachute deployment[?] (See Table 6 for a comparison). Components like an IMU, barometer, and GPS were used in this computer as they have useful applications in amateur rocketry like recovery, data collection, location tracking, thrust vector control (TVC), and motor characterization. These

sensors are also often found on commercial computers. After going through the process of selecting a specific sensor model, the MPU-6050[?] was chosen as the IMU, the MPL3115[?] as the barometer, and the L76-M33 as the GPS[?]. See Section II for the component selection and Section V for the component comparison criteria. Besides the sensors, the computer also includes the ability to control 3 servos for TVC and 6 igniter channels for state separation and recovery. The whole system was designed around a popular microcontroller platform, and after reviewing multiple options, the Arduino was chosen. Specifically the Arduino MEGA and its ATmega2560 microcontroller[?] were used. See Table 1 and Section II.A for a comparison on different microcontrollers. Everything is included on a custom circuit board designed to have a similar footprint to the Arduino MEGA. This computer also supports the standard Arduino Integrated Development Environment (IDE) and comes with an extension of the Arduino Hardware Abstraction Layer (HAL) to offer support for the specific components included in the system.

II. Design Criteria

When creating the flight computer, there are several factors that influence the overall design. This computer is intended to be used as an extensible educational project, so it is important to choose components with an existing user base. These components are usually found on breakout boards that can be easily purchased online and are often used in other projects. With many user examples, there are often reference schematics we can use to design the overall computer's schematic. This also simplifies the software design process as there are likely libraries to reference. SparkFun and Adafruit are manufacturers that produce breakout boards for sensors and other components[? ?]. Since users will often use these in future projects, this allows users to familiarize themselves with these components and then apply them to other projects in the future.

A. Processor and Development Board

The first part of designing the flight computer is deciding which microcontroller ecosystem to base the computer on. Two popular microcontroller platforms are the STM32 and the Arduino. While the STM32's 32-bit architecture, higher performance, and advanced IO is attractive[?], it may be over complicated for beginners and has a development environment that is more suited for experienced engineers. Its STM32CubeIDE offers many advanced features like code generation and peripheral configuration[?]. Arduino's IDE is significantly more user friendly as it uses standard peripheral configurations and handles much of the configuration itself using its HAL and purpose-built language[? ?]. With thousands of amateur projects based around the Arduino, it is the optimal choice for an educational project[?].

Arduino.cc offers several options for development boards, all of which are open source. Three of their most popular boards include the Micro[?], UNO[?], and the MEGA[?]. The boards are largely similar, but most of their differences come in their available IO and form factor. Comparing the 3 options (See Table 1), the MEGA appears to be the best-suited for the computer. It has the most SRAM and IO out of the 3 options. With this computer utilizing multiple sensors, the number of IO pins and the amount of SRAM is important.

Table 1 Arduino Development Board Comparison

Name*	MCU	SRAM	IO	UART	SPI	I2C
Uno[?]	ATmega328P[?]	2Kb	14 Digital, 6 Analog	1	1	1
Micro[?]	ATmega32u4[?]	2.5Kb	20 Digital, 12 Analog	1	1	1
Mega[?]	ATmega2560[?]	8Kb	54 Digital, 16 Analog	4	1	1

B. IMU

In rocket flight computers, it may be necessary to include a sensor to track the acceleration and orientation of the rocket. This is a necessary function when running TVC on a rocket as the orientation feedback can be used to correctly aim the rocket. This data can provide the user with a valuable insight into how their rocket is moving while it is in flight. The accelerometer can be used for motor characterization and can tell the user the maximum force the rocket is experiencing. Gyroscopes are used for tracking the orientation of a device and its angular velocity. These sensors

*Clock speed, EEPROM, and SRAM size were not compared in this table as all 3 processors run at 16MHz, have 256Kb of flash, and 4Kb of EEPROM[? ? ?].

are commonly used in rocket flight computers to track the orientation of the rocket in flight. Usually, accelerometers, gyroscopes, and magnetometers are included on a single chip called an IMU. Magnetometers can be used to find the orientation of the rocket relative to Earth's atmosphere.

In this computer, the MPU-6050 IMU was chosen for its average accuracy[?] and its availability as a common SparkFun breakout board[?].

Table 2 Comparison of Common IMUs

Name	Manufacturer	Breakout	Accuracy	Degrees of Freedom*
MPU-6050[?]	InvenSense	Sparkfun[?]	$\pm 16g$, $\pm 2000\text{deg/sec}$, $\pm 1\%\text{drift}$	6
MPU-9250	InvenSense	Sparkfun[?]	$\pm 16g$, $\pm 2000\text{deg/sec}$, $\pm 1\%\text{drift}$	9
ADXL375[?]	Analog Devices	Analog Devices[?]	$\pm 200g$	3
BMX160	Bosch	dfrobot[?]	$\pm 16g$, $\pm 2000\text{deg/sec}$, $\pm 1\%\text{drift}$	9

C. Barometer

In a flight computer, barometers are used to track the atmospheric pressure which can be used to determine altitude. Pressure is inversely proportional to altitude referenced to sea level. So as the rocket reaches a higher altitude, the pressure decreases. A lookup table can be used to estimate altitude above or below sea level at a given pressure. Users can use this data to track the max altitude the rocket reaches in flight. Common computers also use this to determine when to deploy the recovery parachute. In this computer, activation altitude and charge ignition can be customized.

In this computer, the MPL3115 was chosen specifically for its included altimeter module[?]. It is also available as a breakout board[?]. This allows it to calculate the altitude without external processing or calibration.

Table 3 Comparison of Common Barometers

Name	Manufacturer	Breakout	Accuracy (P)	Accuracy (m)	Features
MS5607[?]	TE Connectivity	Parallax[?]	1	0.2	Altimeter
MPL3115[?]	NXP	Adafruit[?]	1.5	0.3	Altimeter and temperature
BMP280[?]	Bosch	Adafruit[?]	1	1	N/A

D. GPS

GPS devices are uncommon in most entry level computers and usually unnecessary unless the rocket is expected to land at a large distance away from the launch point. GPS can be used to track the rocket's location on Earth using satellites. This is useful if the user wants to find the landing location or how far the rocket travelled over the ground. In this computer, the GPS is only enabled once the rocket is no longer in motion. GPS devices are usually limited in their maximum acceleration[?] to avoid the use in missiles or weaponry. In this computer, the L76-M33 GPS was chosen for its quick acquisition time and its ability to use multiple GPS networks simultaneously.

Table 4 Comparison of GPS

Name	Manufacturer	Breakout	GNSS	Aquisition (s)
MAX-8Q[?]	u-blox	No	GLONASS, GPS, and QZSS	30
L76-M33[?]	Quectel	No	GPS, GLONASS, BeiDou, Galileo and QZSS	15

*6 degrees of freedom corresponds to a 3 axis gyroscope and a 3 axis accelerometer. 3 degrees of freedom refers to an accelerometer only in this chart. 9 degrees of freedom has the same as 6 degrees of freedom but also includes a digital compass/magnetometer.

E. Storage

Many flight computers do not include the ability to log flight data. Logging flight data is essential if a user wants to review any sensor data after the flight is completed. This allows the user to view information like the maximum altitude the rocket has reached or its maximum acceleration. The data can also be plotted for more insight into how the rocket behaved over time. To record the data, both a flash chip and SD card are used. The flash chip records the data in real time and the SD card is used to dump all flight data after the rocket has finished flying in order to transfer data to a laptop or other computer. A standard SD card can be used in this computer. The flash used is a W25Q64 64Mb SPI flash chip[?].

F. Telemetry

In order to provide the user with real-time data and control, it is necessary to include a form of telemetry. Common communication methods found in commercial flight computers include Bluetooth (BT), 900Mhz/LoRa, and 70cm HAM (See Table 6). 70cm HAM requires a specific license[? ?], so it will not be considered as since this is computer is intended for beginners and amateurs. Bluetooth has a higher bandwidth but a shorter range than LoRa, but as we are not streaming large amounts of data, we will use LoRa for its range. In this system will will use the EBYTE E220[?].

III. Design Process

To make the computer more compact and replaceable, the entire flight computer was created on a single printed circuit board in the EasyEDA EDA software. EasyEDA allows a web-based environment for users to create their own schematics and PCBs. It is also user friendly and allows for collaboration so users can recreate the computer if they wish[?]. In this design, many of the schematics for the existing breakout boards of these components were used as reference when designing the computer. The Arduino MEGA reference schematic was also used to duplicate all of the same pin assignments from the original board and insure that the ATmega2560 functions the same as if it were in the MEGA board. This way, the software that is written for this computer will also work if a regular Arduino MEGA is connected to breakout boards.

A. Microcontroller

Since this computer is essentially an Arduino reference board with peripherals that are built directly on to the board, we will use the same ATmega2560 microcontroller and ATmega16U2 serial processor. We also use Arduino's recommended designs for the power supply, reset switch, clock, and USB port[?]. The ATmega operates at a 5V logic level but much of the sensors that are used operate at 3V so logic level conversion will be required (See Fig. 5).

Table 5 Digital I/O Pin Assignment for Peripherals

Component	GPIO	Name	Component	GPIO	Name
Flash SD	PB1	SCK	GPS	PD2	UART1 RX
Flash SD	PB2	MOSI	GPS	PD3	UART1 RX
Flash SD	PB3	MISO	IMU + Barometer	PD0	SCL
Buzzer	PA6	28	IMU + Barometer	PD1	SDA
Flash CS	PC7	30	SD CS	PC6	31
Gyro Rail	PC5	32	Baro Rail	PC4	33
Buzzer Rail	PC3	34	LoRa CS	PC2	35
Servo1	PA0	22	Servo2	PA1	23
Servo3	PA2	24	Servo4	PA3	25
Servo5	PA4	26	Servo6	PA5	27

B. Sensors

For both the MPU-6050 IMU and MPL3115 Barometer, the schematics from the breakout boards were replicated in the computer[? ?]. The included regulators on the boards were not included as the logic level conversion is handled by another component. Other than the main chip, both breakout boards contain mostly decoupling capacitors and pull-up resistors for the I2C bus.

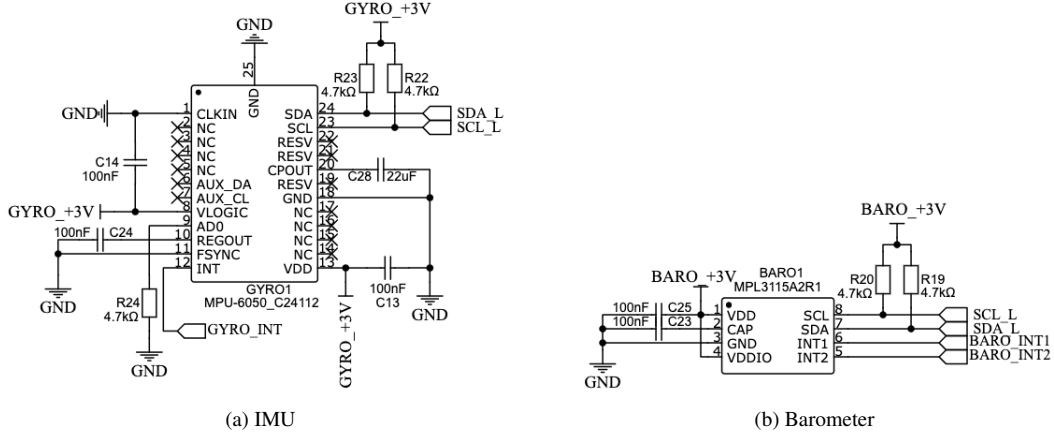


Fig. 1 IMU and Barometer Schematics

The L76 GPS does not have a breakout board but is instead a SoC that contains all the necessary components to perform its functions[? ?]. It only requires decoupling capacitors and a connection to reset, which in this case is pulled high. An LED is also attached for the PPS indicator.

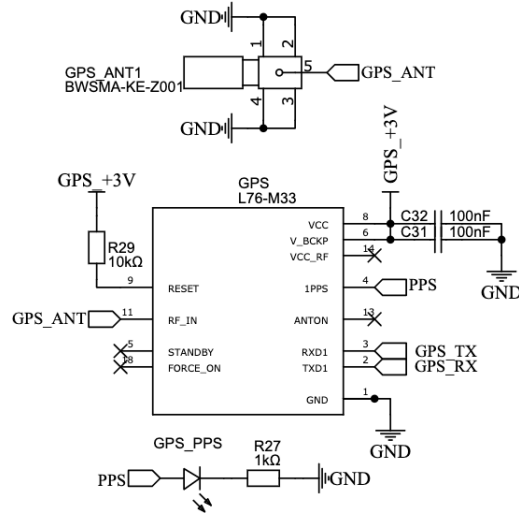


Fig. 2 GPS Schematic

C. Servos and Igniters

To control charges or igniters, it is not necessary for a dedicated IC. Actuation of these can usually be achieved by a simple arrangement of components to control the high current flow (See Fig. 3). In this case, we are using a MOSFET to act as a switch to connect the charge to ground. The MOSFET does require a resistor in series with its gate to drop the voltage. It also requires a pull-down resistor to ground as MOSFETs have a small capacitance between its drain and gate, and this needs to be dissipated[?].

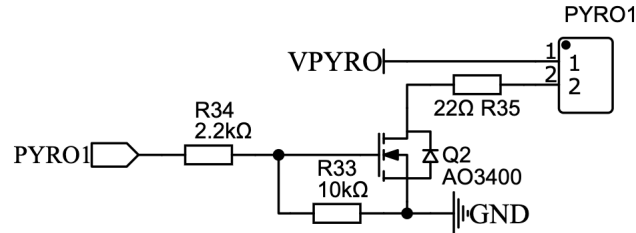


Fig. 3 Charge Actuator Schematic

D. Load Switches

Since this computer has a number of external sensors and peripherals, depending on the application and user requirements, some may not be used. For example, the GPS is only used to track the location of the rocket once it has landed. It is not necessary for the GPS to operate in flight due to government limitations[?], so it is powered off in flight. Each sensor IC is connected to an individual load switch responsible for disconnecting the IC's power. By default, the sensors are disconnected from power unless the user initializes them. This saves power and reduces the chance for errors and unexpected behavior to occur.

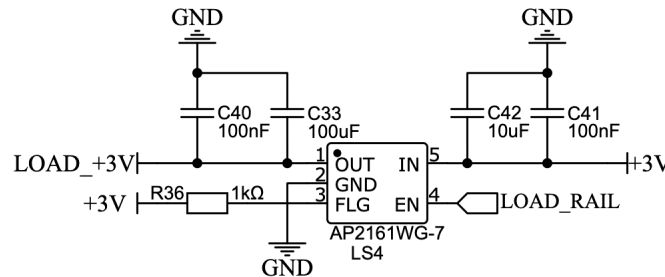


Fig. 4 Load Switch Schematic

E. Logic Level Conversion

To avoid damaging the components, the 5V logic level from the ATmega2560[?] needs to be converted to the 3V that the sensors use. Some of these devices are tolerant of 5V or 3V, but to be within the recommended range, the levels are going to be shifted to 3V in this computer. This is done with a TXS0108 chip[?]. Decoupling capacitors are placed on both the 3V and 5V rails. The output enable is pulled high through a jumper and pulled low when the jumper is disconnected.

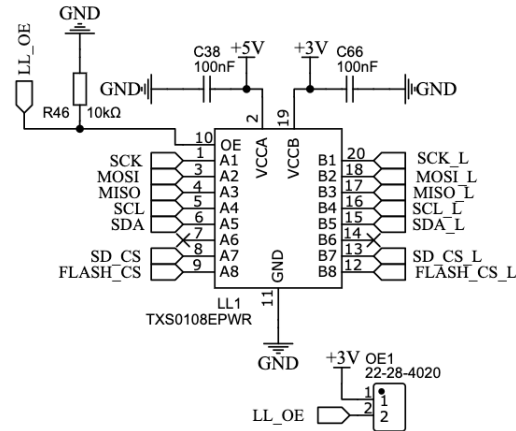


Fig. 5 Logic Level Converter Schematic

F. Storage and Telemetry

Many SD cards use SPI to communicate by default, so nothing needs to be added other than the card holder. The SPI flash chip only requires the SPI connection, decoupling capacitors, and the hold and write-protect signals[?]. The E220 Lora module also uses the same SPI communication as the flash chip[?].

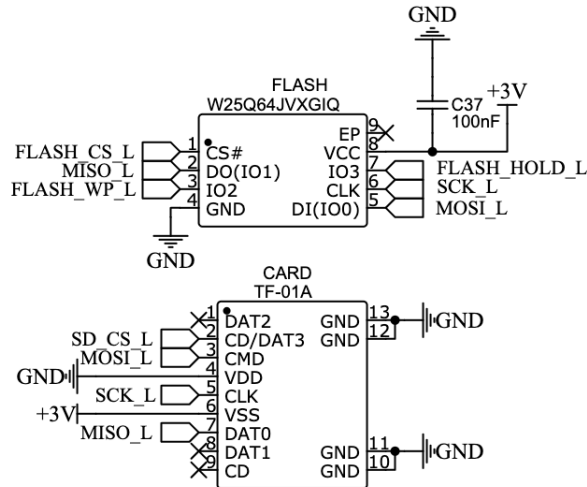


Fig. 6 Storage Schematic

G. Safety

To reduce the chances for accidents to occur, there are two switch terminals included on the board. One terminal is for the main power switch that disconnects power to the entire computer. The other terminal is for arming the ignition charges. This helps prevent unintended ignition do to software or hardware errors.

IV. Software

In order to read the sensor data and perform functions, it is necessary for the flight computer to include a software package. Since this computer is designed to use standard Arduino code, we can reuse much of the Arduino HAL. The Arduino Hardware Abstraction Layer (HAL) is used as an interface between the user and the code necessary to operate the microcontroller and to interact with the hardware registers for IO control. We can build off this Arduino HAL and create our own system. Instead of including the "Arduino.h" header file when writing code, we can include our custom header file. In our HAL, we have access to the standard functions and a wrapper around the libraries for the sensors. As an example, we will have dedicated functions to pull data from each sensor without having to understand which IO port and protocol that is required for that device. The libraries and protocol information to enable communication with each component are included in their datasheets. These protocols are then implemented in the new HAL. Initialization is handled just like the original Arduino HAL. Components can be selectively enabled and disabled by controlling their power rails.

V. Comparison to Commercial Computers

Using EasyEDA's bill-of-materials (BOM) tool, then total cost for this computer comes out to around \$75*. This is priced significantly lower than the basic flight computers that are compared in Fig. 6 and costs 600% less than a higher-end computer.

Table 6 Hardware Comparison

Name	Price (est)	MCU	Charges	Sensors	Telemetry	Other
Signal[?]	\$349	ATSAMD21	3	IMU, Barometer	BT [†]	TVC
AVA[?]	N/A [‡]	3 Independent	6	IMU, GPS, Barometer	BT + LoRa	TVC
EasyMini[?]	\$102[?]	LPC1114	2	Barometer	N/A	Buzzer
EasyMega[?]	\$358[?]	STM32L151	4	IMU, Barometer	N/A	Buzzer
TeleMetrum[?]	\$382[?]	STM32L151	2	IMU, GPS, Barometer	70cm ham	Buzzer
TeleMega[?]	\$508[?]	STM32L151	4	IMU, GPS, Barometer	70cm ham	TVC, Buzzer
This computer	\$75*	ATmega2560	4	IMU, GPS, Barometer	LoRa	TVC, Buzzer

When looking at the above chart, it can be seen that this computer is comparable to the BPS.space Signal and TeleMega, but at a significantly reduced price. The BPS.space Signal offers TVC and has some software customization but has a more complicated software package. This computer is priced in the same range as a simple computer that only offers a barometer for recovery charge deployment.

VI. Conclusion

In this paper, we went through the process of selecting components for creating a flight computer that is user-friendly and based on the Arduino platform. We were able to design a device that had more features than expensive computers while staying in the price range of budget computers with minimal features. With this computer, there are many possibilities that can be explored in the future. With the basic hardware capabilities, many software features can be added to enable more complex control over a rocket.

*The price varies depending on component prices and minimum order quantity. This price was calculated using the individual unit price instead.

[†]Bluetooth is shortened to "BT"

[‡]This flight computer is not in production

References

- [] BPS.space, “BPS.space Signal V2 Product Description,” , 2023. URL <https://bps.space/products/signal-r2>.
- [] Rockets, A., “TeleMetrum Product Description,” , 2023. URL <https://www.apogeerockets.com/Electronics-Payloads/Altimeters/TeleMetrum>.
- [] Robotics, A., “Easy Mini Product Description,” , 2022. URL <https://www.apogeerockets.com/Electronics-Payloads/Dual-Deployment/EasyMini>.
- [] *MPU-6000 and MPU-6050 Product Specification*, InvenSense, 8 2013. Revision 3.4.
- [] *MPL3115A2 I2C precision pressure sensor with altimetry*, NXP, 12 2018. Revision 8.
- [] Arduino, “Arduino Mega 2560 Rev 3 Overview,” , 2023. URL <https://store.arduino.cc/products/arduino-mega-2560-rev3>.
- [] Sparkfun, “SparkFun Triple Axis Accelerometer and Gyro Breakout,” , 2023. URL <https://www.sparkfun.com/products/11028>.
- [] Adafruit, “Adafruit BMP280 I2C or SPI Barometric Pressure Altitude Sensor - STEMMA QT,” , 2023. URL <https://www.adafruit.com/product/1893>.
- [] STMicroelectronics, “STM32 Microcontrollers,” , 2023. URL <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>.
- [] STMicroelectronics, “STM32CubeIDE,” , 2023. URL <https://www.st.com/en/development-tools/stm32cubeide.html>.
- [] Arduino.cc, “Getting Started with Arduino IDE 2.0,” , 2023. URL <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2>.
- [] Arduino.cc, “Arduino Reference,” , 2023. URL <https://www.arduino.cc/reference/en/>.
- [] Arduino.cc, “What is Arduino?” , 2018. URL <https://www.arduino.cc/en/Guide/Introduction>.
- [] Arduino, “Arduino Micro Overview,” , 2023. URL <https://store-usa.arduino.cc/collections/boards/products/arduino-micro>.
- [] Arduino, “Arduino UNO Rev 3 Overview,” , 2023. URL <https://store.arduino.cc/products/arduino-uno-rev3>.
- [] *ATmega328P Datasheet*, Atmel, 1 2015. 7810D-AVR-01/15.
- [] *ATmega16U4/ATmega32U4 Datasheet*, Atmel, 4 2016. Atmel-7766J-USB-ATmega16U4/32U4-Datasheet04/2016.
- [] *Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V Datasheet*, Atmel, 2 2014. 2549Q-AVR-02/2014.
- [] Sparkfun, “SparkFun Triple Axis Accelerometer and Gyro Breakout,” , 2023. URL <https://www.sparkfun.com/products/11028>.
- [] *ADXL375 3-Axis, ± 200 g Digital MEMS Accelerometer*, Analog Devices, 7 2018. Revision 0.
- [] Devices, A., “ADXL375 3-Axis, ± 200 g Digital MEMS Accelerometer,” , 2023. URL <https://www.analog.com/en/products/adxl375.html#product-overview>.
- [] dfrobot, “Fermion: BMX160 9-axis Sensor,” , 2023. URL <https://www.dfrobot.com/product-2141.html>.
- [] Adafruit, “Adafruit MPL3115A2 - I2C Barometric Pressure/Altitude/Temperature Sensor,” , 2023. URL <https://www.adafruit.com/product/1893>.
- [] *MS5607-02BA03 Barometric Pressure Sensor, with stainless steel cap*, TE Connectivity, 11 2020. TE CONNECTIVITY SENSORS ///MS5607-02BA03.
- [] Parallax, “Altimeter Module MS5607,” , 2023. URL <https://www.parallax.com/product/altimeter-module-ms5607/>.
- [] *BMP280 Digital Pressure Sensor*, Bosch, 5 2015. Revision 1.14.
- [] *Quectel L76 Compact GNSS Module*, Quectel, 5 2018. Revision 1.

- [] u-blox, “MAX-8 series u-blox 8 GNSS modules,” , 2023. URL <https://www.u-blox.com/en/product/max-8-series>.
- [] *W25Q64JV 3V 64M-BIT SERIAL FLASH MEMORY WITH DUAL, QUAD SPI*, Winbond, 3 2021. Revision K.
- [] Rockets, A., “Easy Mega Product Description,” , 2023. URL <https://www.apogeerockets.com/Electronics-Payloads/Altimeters/TeleMega>.
- [] *E220-900T22D*, EBYTE, 3 2022. E220-900T22D.
- [] EasyEDA, “EasyEDA,” , 2023. URL <https://easyeda.com>.
- [] *Arduino Mega 2560 Rev 3 Product Reference Manual*, Arduino, 2 2023. SKU A000067.
- [] *AO3400 30V N-Channel MOSFET*, Alpha and Omega Semiconductors, 12 2011. Revision 8.
- [] *TXS0108E8-Bit Bi-directional, Level-Shifting, VoltageTranslator for Open-DrainandPush-Pull Applications*, Texas Instruments, 12 2007. REVISED MAY 2020.
- [] BPS.space, “The AVA Flight Computer,” , 2020. URL <https://www.youtube.com/watch?v=qaWvCy2DRSA>.
- [] AltusMetrum, “Easy Mini Product Description,” , 2022. URL <https://altusmetrum.org/EasyMini/>.
- [] AltusMetrum, “Easy Mega Product Description,” , 2021. URL <https://altusmetrum.org/EasyMega/>.
- [] Robotics, A., “Easy Mega Product Description,” , 2021. URL <https://www.apogeerockets.com/Electronics-Payloads/Dual-Deployment/EasyMega>.
- [] AltusMetrum, “TeleMetrum Product Description,” , 2020. URL <https://altusmetrum.org/TeleMetrum/>.
- [] AltusMetrum, “TeleMega Product Description,” , 2020. URL <https://altusmetrum.org/TeleMega/>.