Component Analysis

Year: 2022 Semester: Fall Team: 8 Project: Hermes

Creation Date: August 30, 2022 Last Modified: September 17, 2022

Member 1: Michael Langford Email: [mjlangfo@purdue.edu](mailto:mjlangfo@purdue.edu)

Member 2: Jonathan Wosiak Email: [jwosiak@purdue.edu](mailto:jwosiak@purdue.edu)

Member 3: Santiago J. Garcia Delgado Email: [garciads@purdue.edu](mailto:garciads@purdue.edu)

Member 4: Owen Mandel Email: [omandel@purdue.edu](mailto:omandel@purdue.edu)

Assignment Evaluation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Score (0-5)** | **Weight** | **Points** | **Notes** |
| **Assignment-Specific Items** | | | | |
| **Analysis of Component 1** |  | x2 |  |  |
| **Analysis of Component 2** |  | x2 |  |  |
| **Analysis of Component 3** |  | x2 |  |  |
| **Bill of Materials** |  | x6 |  |  |
| **Writing-Specific Items** | | | | |
| **Spelling and Grammar** |  | x2 |  |  |
| **Formatting and Citations** |  | x1 |  |  |
| **Figures and Graphs** |  | x2 |  |  |
| **Technical Writing Style** |  | x3 |  |  |
| **Total Score** |  | | |  |

5: Excellent 4: Good 3: Acceptable 2: Poor 1: Very Poor 0: Not attempted

General Comments:

1.0 Component Analysis:

The major hardware components of our Hermes drone implementation are the Microcontroller, the IMU, the radio receiver, motor controller, Raspberry Pi Zero, camera, and motors. The microcontroller acts as the flight controller for the overall vehicle, interfacing with sensors, processing data, and controlling motors. The IMU will detect rotational and acceleration changes to keep the drone stable and flying. The radio receiver will be used to control the vehicle when it is not in autonomous mode. The Pi Zero and camera will work together to identify a target. The motor controller will take dictated motor speed commands and drive the motors. Lastly, the motors will be responsible for the movement of the drone.

1.1 Analysis of Component 1: Microcontroller

The microcontroller has the role of flight controller, meaning it will need to manage communications between all our subsystems. To ensure that it can do this it will need 3 different types of connections, SPI, I2C, UART. The SPI protocol will be responsible for communicating with the IMU to receive rotational and acceleration data. The I2C protocol LIDAR sensor which will be used to detect the proximity of objects to our drone, will communicate that information back to our microcontroller. The UART protocol will be used to communicate between the radio receiver and the microcontroller.

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | STM32F446RE | STM32F411CE | STM32F417VE |
| Data Bus Width | 8-bit, 16-bit | 8-bit | 8-bit, 14-bit |
| Clock speed | 180 MHz | 100MHz | 168MHz |
| # of pins | 64 | 48 | 100 |
| # SPI connections | 4 | 5 | 3 |
| # UART connections | 2 | 0 | 2 |
| # I2C connections | 3 | 3 | 3 |
| Timers | 17 | 11 | 17 |
| Flash Memory | 512 Kbytes | 512 Kbytes | 512 Kbytes |
| Operating Voltage | 1.8 to 3.6 V | 1.7 to 3.6 V | 1.8 to 3.6 V |
|  | Chosen |  |  |

Since our flight controller needed all 3 of those protocols and we knew that for this type of complicated processing the ARM system would be best suited for the task. We decided that for our microcontroller we would need an STM32F4, but we weren’t sure which model would be best suited for our purposes. Since we are building a drone Clock speed, timers and the data bus width were the most important factors in our decision. Given this we found that STM32F446RE provided us with the most important features for our drone to operate efficiently.

1.1 Analysis of Component 2: IMU

The IMU detects rotational and acceleration changes for our drone. With the data we are receiving from the IMU we will be able to calculate the yaw, pitch and roll of the overall drone and use that data to stabilize the drone. We will be using SPI to interface with the IMU. Much of the decision relied on the availability of components

|  |  |  |
| --- | --- | --- |
| Feature | **LSM6DSOWTR** | ICM20948 |
| Package Size | 2.5 x 3 x 0.38 mm | 3 x 3 x 1 mm |
| Gyro Full Scale (dps) | ±250, ±500, ±1000, ±2000 | ±250, ±500, ±1000, ±2000 |
| Accel Full Scale range | ±2, ±4, ±8, ±16 | ±2, ±4, ±8, ±16 |
| Digital Output | I2C or SPI | I2C or SPI |
| Operating Voltage | 1.71 - 3.6 | 1.71 - 3.6 |
|  | Chosen |  |

Both IMUs’ are great for the role that they will serve on our drone. We have yet to decide on which component to utilize, however the ICM20948 has been recorded to have less noise, this makes it more desirable for us. However, much of the decision relied on the availability of components and it appears that at the moment the LSM6DSOWTR is much more available than any other IMU.

1.1 Analysis of Component 3: Radio Receiver

The radio receiver is meant to transmit user input to the microcontroller in order to non-autonomously fly the drone. This is the only real interaction the user will have with the drone in order to fly it. The remote control will have two joysticks that will represent the data for side-to-side and up and down movement. This will then be transmitted to the radio receiver on the drone where it will connect to the microcontroller via UART.

|  |  |  |
| --- | --- | --- |
| Feature | **BetaFPV ELRS Lite** | RadioMasterR834 V2 |
| Input Voltage | 5V | 4.5 - 6V |
| Frequency Bands | 2.4GHz ISM | 2400-2483.5MHz |
| Size | 10x10 mm | 17x11 mm |
| Weight | 0.47g or 0.53g (with SMD ceramic antenna) | 2 g |
|  | Chosen |  |

The choice was very easy between BetaFPV and the RadioMaster for the simple reason of the size and of communication. First it is apparent that BetaFPV is much smaller than the RadioMaster. And then the communication of the BetaFPV is through UART, while the RadioMaster uses PWM.

1.1 Analysis of Component 4: Motors

The motors will be used in order to make the drone capable of flight. Four motors are going to be mounted to the corners of the frame of the drone in order to have full control over the yaw, pitch and roll of the vehicle as a whole. These four motors will be connected to an electric speed controller (ESC) and will then connect to the microcontroller via DSHOT, a digital ESC communication protocol.

2.0 Sources Cited:

1. STMicroelectronics. STM32F415xx STM32F417xx (2020). Accessed: Sep. 17 2022. [Online] Available: <https://www.st.com/resource/en/datasheet/stm32f415rg.pdf>
2. STMicroelectronics. STM32F411xC STM32F411xE (2017). Accessed: Sep. 17 2022. [Online] Available: <https://www.st.com/resource/en/datasheet/stm32f411ce.pdf>
3. STMicroelectronics. STM32F446xC/E (2021). Accessed: Sep. 17 2022. [Online]. Available: <https://www.st.com/resource/en/datasheet/stm32f446mc.pdf>
4. STMicroelectronics. LSM6DS3TR-C(2021). Accessed: Sep. 17 2022. [Online]. Available: <https://www.mouser.com/datasheet/2/389/lsm6ds3tr_c-1761429.pdf>
5. TDK. ICM-20948 . Accessed: Sep. 17 2022. [Online]. Available: <https://invensense.tdk.com/wp-content/uploads/2021/10/DS-000189-ICM-20948-v1.5.pdf>
6. RaceDayQuads. BetaFPV ELRS Lite 2.4GHz Receiver - Choose Version. Accessed: Sep. 17 2022. [Online] Available: <https://www.racedayquads.com/products/betafpv-elrs-lite-2-4ghz-receiver>
7. Getfpv. RadioMaster 484 V2 4CH PWM Receiver – D8/D16/SFHSS. Accessed: Sep. 17 2022. [Online] Available: <https://www.getfpv.com/radios/receivers/radiomaster-r84-v2-4ch-pwm-receiver-d8-d16-sfhss.htmlz>