|  |  |
| --- | --- |
| **9V Alkaline Battery (3 Points)** | |
| 1. Must supply voltage that is above the dropout range of the voltage regulator at a current draw of up to 100mA (2pt) | 1. Design a variable load test circuit with the multimeter monitoring the voltage across the supply  2. Tune the load resistance to adjust the load current to 100mA, measure the supply voltage drop and check if it is above the 4.3V margin(regulator voltage plus dropout voltage); |
| 2. The battery needs to support normal operation of the device for at least 1 hour (1pt) | 1. Connect a load to the battery that draws consistent current;  2. Check if the voltage of the battery is above 5V after 1 hour. |
| **Voltage Regulator (5 Points)** | |
| 1. Output voltage must be regulated to +3.3V ± 5% at a range of current draw (up to 100mA) as required by the modules (5pts) | 1. Connect the multimeter across the voltage regulator;  2. Check if the voltage drop between the input and output of the voltage regulator matches specification;  3. Use multimeter to track if the output voltage is stable;  4. Adjust the load current to 15mA and 30mA, respectively, repeat the above process. |

Table 5: Power Supply Unit RV Table

|  |  |
| --- | --- |
| **Status LED (0 Points)** | |
| 1. At forward current of 10mA, the LED should emit red light and visible at direct viewing angle | 1. Connect the LED in series with a 330 Ohm resistor;  2. Use the multimeter to measure current through the LED;  3. At 10mA, make sure the LED light is visible from direct viewing angle |

Table 6: Status LED RV Table

|  |  |
| --- | --- |
| **Sync Button (0 Points)** | |
| 1. Buttons should be debounced and should indicate correct signal transition upon pressed | 1. Connect the button in series with a 330 Ohm resistor;  2. Supply 3.3 V;  2. Use the oscilloscope to monitor the voltage transition across the resistor, make sure the signal is debounced and correspond to correct digital value (digital “High” when pressed and “Low” when released) |

Table 7: Buttons RV Table

|  |  |
| --- | --- |
| **Requirements** | **Verification** |
| **AVR Microcontroller (15 Points)** | |
| 1. Module is capable of running UART at a baud rate of 115200 (3pts) | 1. Program the AVR to send 0x55 repetitively through the UART Tx pin 2. Probe the Tx pin with the oscilloscope 3. Verify the presence of a square wave with each pulse lasting ~8.68µs |
| 1. Device is capable of writing text files to the microSD card that can be read from a PC (7pts) | 1. Write a simple test program onto the AVR that uses the roland-riegel MMC/SD/SDHC card library to write 16-bit signed integers onto the microSD card 2. Verify on the PC that the number are readable and consistent with what was written |
| 1. Device is capable of monitoring the voltage of the battery when the MOSFET is ON. The reported voltage should be within ± 5%.(2pt) | 1. Get readings from the analog pin 25 that reports the voltage of the battery by using the formula Vbattery = 3\*Vadc;  2. Check if the voltage matches voltmeter readings. |
| 1. Device is capable of receiving data from the IMU via I2c at a minimum bitrate of 5580 bps (calculated requirement for maintaining the target 180 Hz sampling rate) (3pts) | 1. Orient the IMU such that the gyroscope has a constant non-zero reading  2. Pull data from the IMU at 180 Hz for a second  3. Verify that the quantity and validity of data received meets expectations |

Table 8: Microcontroller RV Table

|  |  |
| --- | --- |
| **Bluetooth Module (5pts)** | |
| 1. Module is capable of receiving and transmitting over UART at a baud rate of 115200 (4pts) | 1. Program an Arduino to start serial communication at the baud rate of 115200 2. Use the Arduino to program the Bluetooth module to operate at baud rate 115200 3. Connect the Bluetooth module to an Arduino (Tx->Rx, Rx->Tx) 4. Short the Tx and Rx pins 5. Pair computer with Bluetooth module 6. Use a terminal program (eg. puTTY), connect to the correct serial line at baud rate 115200 7. Verify that what’s typed into the terminal is echoed back. If so, the loopback test is successful |
| 2. Two of such modules are capable of performing at the speed mentioned above when placed 2 ± 5% meters apart (1pt) | 1. Repeat steps 1-6 from the verification above 2. Pre-populate a string field of length 115200kb 3. Move the HC-05 2 meters away from the testing PC 4. Send the pre-populated string and check for accurate echo response |

Table 9: Bluetooth RV Table

|  |  |
| --- | --- |
| **Accelerometer (5 Points)** | |
| 1. Acceleration output accurate up to 5%. | 1. Connect the accelerometer to microcontroller with test code on it. The test code will collect the acceleration data from the accelerometer.  2. Turn on the test circuitry and use one hand to hold the accelerometer about 6 inches above the other hand. Based on the accelerometer the x axis should be facing upward.  3. Drop the accelerometer from one hand to the other hand. Try to drop is so the accelerometer drops straight down and does not rotate during the fall.  4. Repeat step 2 and step 3 with the y axis and the z axis.  5. Port the collected data and graph it. Recommend using MATLAB but other scripting languages and environments could work too.  6. Plot the data and verify that when the accelerometer is held, the reading for the tested axis should be 1 g. Then the data should drop to 0 g when the accelerometer is falling. Both readings should be within the specified error margin. |
| 3. Samples at a rate of at least 180 Hz in order to give enough resolution to represent movement | 1. Connect the accelerometer to the microprocessor and load a test code that will collect and log the data from the accelerometer  2. Obtain a stopwatch to record the time data is collected. Collect data for 10 seconds.  3. Analyze the data collected to ensure that there are enough entries to meet the requirement of 180 Hz (for 10 seconds example there should be at least 1800 entries) |
| **Gyroscope (5 Points)** | |
| 1. Accurate up to ± 7 degrees in order to assure accuracy in data | 1. Connect the gyroscope to a microcontroller that has test code to collect data from the gyroscope.  2. Turn on the test circuitry with the gyroscope lying flat.  3. Rotate the x axis (roll) 90 degrees and then rotate back to initial position.  4. Repeat step 3 for the y axis (pitch).  5. Port the collected data and graph it. Recommend using MATLAB but other scripting languages and environments could work too.  6. Plot the data and verify that the axis being tested reaches 90 degrees or is within the error margin. The data should also reach 0 degrees when rotated back to the initial position. If the data collected is the raw angular velocity then, integrate it before plotting. |
| 2. Samples at a rate of at least 180 Hz in order to give enough resolution to represent movement | 1. Connect the gyroscope to the microprocessor and load a test code that will collect and log the data from the gyroscope  2. Obtain a stopwatch to record the time data is collected. Collect data for 20 seconds.  3. Analyze the data collected to ensure that there are enough entries to meet the requirement of 180 Hz (for 20 seconds example there should be at least 3600 entries) |

Table 10: IMU RV Table

|  |  |
| --- | --- |
| **Force Sensitive Resistor and Low-pass Filter (12 Points)** | |
| 1. Pressure measurement range needs to be from 1psi to at least 6.409psi (Equivalent: force measurement range needs to be between 14-90N);  The measurement needs to be within ±10% of the theoretical value at the two boundary values. (4pts)  \*see derivation of the values used here in the calculation and simulation section | 1. Place the FSR on a flat surface and connect the FSR in series with a 20 kOhm reference resistor Rref;  2. Supply 3.3V voltage across the two resistors;  3. Place a 1.4kg weight on top of the FSR;  4. After the readings stabilize, measure the voltage drop across the reference resistor, Vref;  5. Find VFSR by using the relation VFSR = 3.3V - Vref and then find the current across the components I = Vref / Rref;  6. Use the voltage and current found to calculate FSR resistance, compare with theoretical value and check if it is within 10% error margin;  7. If the value does not match the standards, change the weight until the value meets the requirement, record the adjusted value and check if it is among a reasonable range;  8. If the value matches the standards, move on to replace the 1.4kg weight with a 9.2kg weight and repeat the above steps. |
| 2. Measurement result needs to be accurate within ±10% of the theoretical value (3pts) | 1. Select 5-10 weight values between the two threshold values;  2. Repeat the same steps as specified in part 1 of FSR verification and record the corresponding measurements;  3. Use the values obtained to plot the Pressure vs Force curve, fit a trend line;  3. Check if the trendline matches theoretical curve as shown in Fig. 4 and the points have less than ±10% error. |
| 3. Must draw current less than 1uA at resting state (2pt) | 1. Supply 3.3V voltage across the resistor;  2. Measure the current drawn from the supply using the multimeter; |
| 4. The low-pass filter must have -3dB frequency of 20Hz ± 10%; (3pt) | 1. Use the function generator to generate a sinusoidal input with amplitude of 50mV at frequency of 5Hz and slowly increase the frequency to 30Hz while using the oscilloscope to monitor the amplitude;  3. When the amplitude dropped to , verify that it is within ± 10% of 20Hz. |

Table 11: FSR RV Table