Appendix A Requirement and Verification Table

1. Control Module

1.1 Microcontroller

Requirement	Verification	Points
1. When in non-auto mode, the microcontroller should be able to adjust the window to different levels using UP and DOWN button. The response time should be less than 2 second. 2. When in auto mode, and the user enter a city that is not in bad weather condition, the microcontroller can get the desired level and adjust the window level to the desired one. The response time should be less than 30 seconds. 3. In auto mode, when user enters a desired level of 4 (which means don't care), the microcontroller should open the window to the level which in long term can drive the room temperature towards the desired one. Meanwhile, we should see that the right dot on seven segment display denoting the Air Conditioner will be on if and only if the window is closed and the room temperature doesn't equal to the desired one. The response time should be less than 30 seconds. 4. If the outside weather condition is bad (rainstorm, tornado, etc.), the window will be closed.	1. Use button to adjust window level from 0 to 3, and back from 3 to 0. Use timer to measure the response time. 2. Find a city not in bad weather condition. And enter the city and desired level of 1 to the webpage, see that the window level will be adjusted to 1. And enter desired level of 0 to the webpage, see that the window level will be adjusted to 0. Use timer to measure the response time. 3. Enter desired level of 4 to the webpage. Enter the desired temperature equals to the outside temperature, see that the microcontroller will adjust the window to level 3. Change the desired temperature to somewhere between the inside room temperature and the outside temperature. See that the microcontroller will adjust the window to level 1 or 2. Change the desired temperature to room temperature. See that the microcontroller will adjust the window to level 0. Check if the right dot on seven segment display is on in appropriate situations. Use timer to measure the response time. 4. Enter a city that is in bad weather condition, and set the desired level to non-zero, see that the window will not be opened. If we set the desired temperature equal to the outside temperature, the window still cannot be opened.	15

1.2 Other Components

Requirement	Verification	Points
1. Temperature Sensor (LM35) (a) The program should calculate the correct temperature based on the voltage output from the temperature sensor. The value should be within an accuracy of 95%.	1. Temperature Sensor (LM35) (a) Compare the calculated temperature with the real temperature read from the room air conditioner.	2
2. Seven Segment Display (ACDA03-41CGKWA-F01) (a) Temperature displayed on the LED must be of the same value as the one read from the temperature sensor and generated by the program. (b) Each segment should be in series with resistor of appropriate resistance to have moderate luminosity.	2. Seven Segment Display (ACDA03-41CGKWA-F01) (a) Write code to read current temperature from the temperature sensor and print it to monitor. Compare the value with the displayed number. (b) Measure the voltage across each segment. The voltage should be around 2.2V	5
3. Switch (a) Should have a pull-up voltage; outputs high when released and low when pressed (b) The program flags, e.g. Auto, Up and Down, should be set to its opposite Boolean value once the according switch is pressed.	3. Switch (a) Use program to print digital reading of the pin which connects to the switch. The output should be 1 when button is released and 0 when pressed. (b) Print flag value before and after the switch is pressed. It should change from TRUE to FALSE or from FALSE to TRUE.	3
4. I^{2}C-compatible Analog-to-Digital Converter (ADS1115) (a) With Adafruit_ADS1115 library, the program can read pin A0-A3 values. (b) The chip can communicate with microcontroller through SDA and SCL successfully.	4. I^{2}C-compatible Analog-to-Digital Converter (ADS1115) (a) Connect a working temperature sensor to each pin. Warm or cool down the sensor. The program should output increasing or decreasing value. (b) Same as (a).	3
5. I/O Expander (PCA9535) (a) The program should set all the 16 bit pins as output. Each bit should output low to turn on LED, and high to turn off LED. (b) The program should have a digit mapping from each output pin to	5. I/O Expander (PCA9535) (a) Write a blinking program to alternate each bit output between high and low. The 7-segment display LED driven by this chip should blink.	5

each segment in the 7-segment display such that it should show the integer value input from the program.	(b) Input the program with values from 0 to 99 and the dots and check if the 7-segment display displays the corresponding values.	

2. Web Module

2.1 Web Server

Requirement	Verification	Points
1. The web server should be able to read following information: city that user lives in, the desired temperature and desired window level. The back end should fetch correct weather information based on the city. Such information should be available in 1 minute: the desired window level, the desired room temperature, whether it is in bad weather condition outside, and the outside temperature. 2. The web server should be able to communicate with the microcontroller through TCP and HTTP. The Web server should be able to handle http request and send back correct information on: the desired window level, the desired room temperature, whether it is in bad weather condition outside, and the outside temperature. 3. The web server should be able to show connection condition. The "connection established" should be updated within 30 seconds, and the "connection lost" should be updated in 3 minutes.	1. Manually enter different values and click on buttons. Use curl command to check if the fetched content is the same as the true weather conditions. 2. Use application called "postman" to send http request, check the reply message is correct. 3. First, check that the web page shows "connection not viable". Use application called "postman" to send http request. Check that the web page shows "connection is viable". Wait for 3 minutes, and see if the web page shows "connection not viable" again.	10

3. Window Module

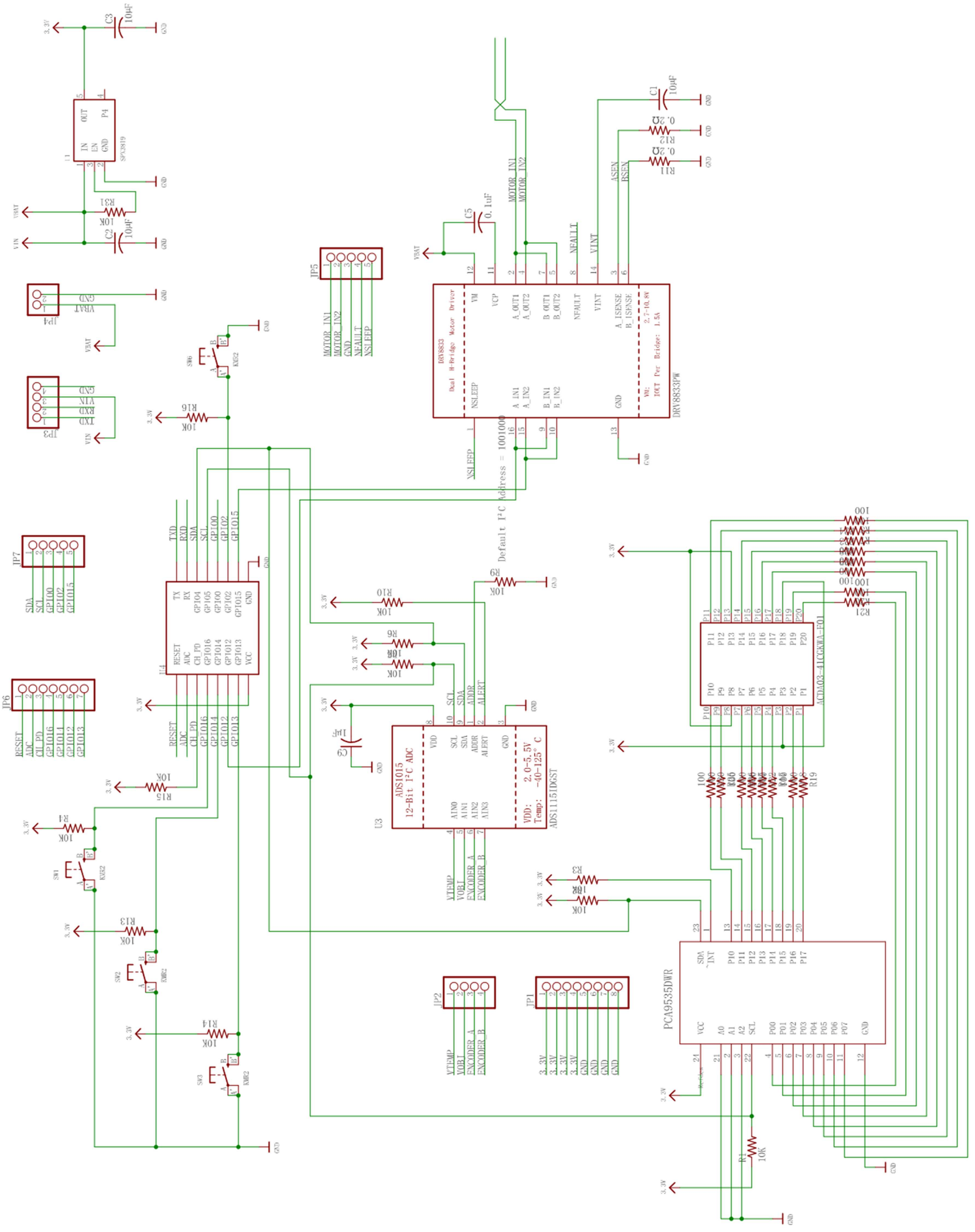
3.1 Infrared Sensor

Requirement	Verification	Points
1. Window stops moving when object is detected, regardless of the way microcontroller was trying to move the window.	1. Simulate situations such that microcontroller will move the window. While the window is moving, place any object between the upper part and bottom part of the window and see if the window stops.	2

3.2 Dual H-Bridge Motor Driver

Requirement	Verification	Points
1. The motor driver can communicate with the microcontroller successfully.	1. Use a small motor for simplicity. Analog Write to each motor pin, the pin should output some significant voltage.	5
2. The motor driver should drive motor in two directions, clockwise and anti-clockwise.	2. Use a small motor for simplicity. Analog Write to motor pin A with significant value, and the other pin B with 0 value. The motor should rotate clockwise, and vice versa.	
3. The motor should be driven in relatively safe and fast speed. The window should be adjusted by one level up/down in 10 seconds.	3. Given the full speed (1024) we use Analog Write, the motor should adjust the window to a level up or down in 10 seconds.	

Appendix B PCB Schematic



SPX3819
DRV8833
Dual H-Bridge Motor Driver
DRV8833PWP
VM: 2.7-10.8V
IOUT Per Bridge: 1.5A
Default I²C Address = 1001000
ADS1015
12-Bit I²C ADC
ADS1115IDGST
VDD: 2.0-5.5V
Temp: -40-125° C
PCA9535DWR
ACDA03-41CGKWA-F01
MOTOR_IN1
MOTOR_IN2
NFAULT
NSLEEP
VINT
ASEN
BSEN
VTEMP
VOBJ
ENCODER_A
ENCODER_B
JP1
JP2
JP3
JP4
JP5
JP6
JP7
SW1
SW2
SW3
SW6
U3
C1
C2
C3
C5
C6
R1
R3
R6
R9
R10
R11
R12
R13
R14
R15
R16
R19
R21
R31
10uF
0.1uF
1uF
0.2Ω
10K
100
3.3V
VBAT
VIN
GND