Åbo Akademi University
Programming Embedded Systems

Project report: Indoor environment control device

Group members: Vu Nguyen (71141)
Gao Yuan (71142)

1. Project overview

1.1. Objective

The aim of the project is to create a small device capable of controlling temperature, light, and humidity in a room. A user can select the temperature, light and humidity level by setting some
parameters to the device via the user interface. The device, in turn, uses those settings by the user to control the temperature, light and humidity level indoor, for example, a room accordingly.

1.2. Background

The device shall have a user interface for displaying temperature, humidity and light levels at all time. In addition, this interface will be used for a user to set temperature and light levels as desired. There shall be push buttons so that the user can select temperature and humidity levels. As a prototype, three LEDs are required to simulate the transducers – that is a heater, a humidifier and a light bulb. Regarding the sensors, the sensors can be analogue sensors or digital sensors as long as they are easy to interface with, support I2C, 1 wire or SPI protocol. Lastly the heart of the device is the platform. It is required to be chosen so that it is affordable, support good core libraries, 8-bit microcontroller based, and has large support or usage from the open source community.

The device should be able to operate in normal temperatures - that is, from 0 to 80 degrees. It does not have be fast since it only interacts with the users when they need to set the temperature, temperature, turn light on or off, and displays these parameters on an interface device. Moreover, it should be hard-coded to keep the room temperatures between 20 degrees to 30 degrees, to keep the humidity levels between 30% and 40%, and to keep light intensity from 445 to 455 lux. Figure 1 shows the system block diagram that gives a big picture of the device.

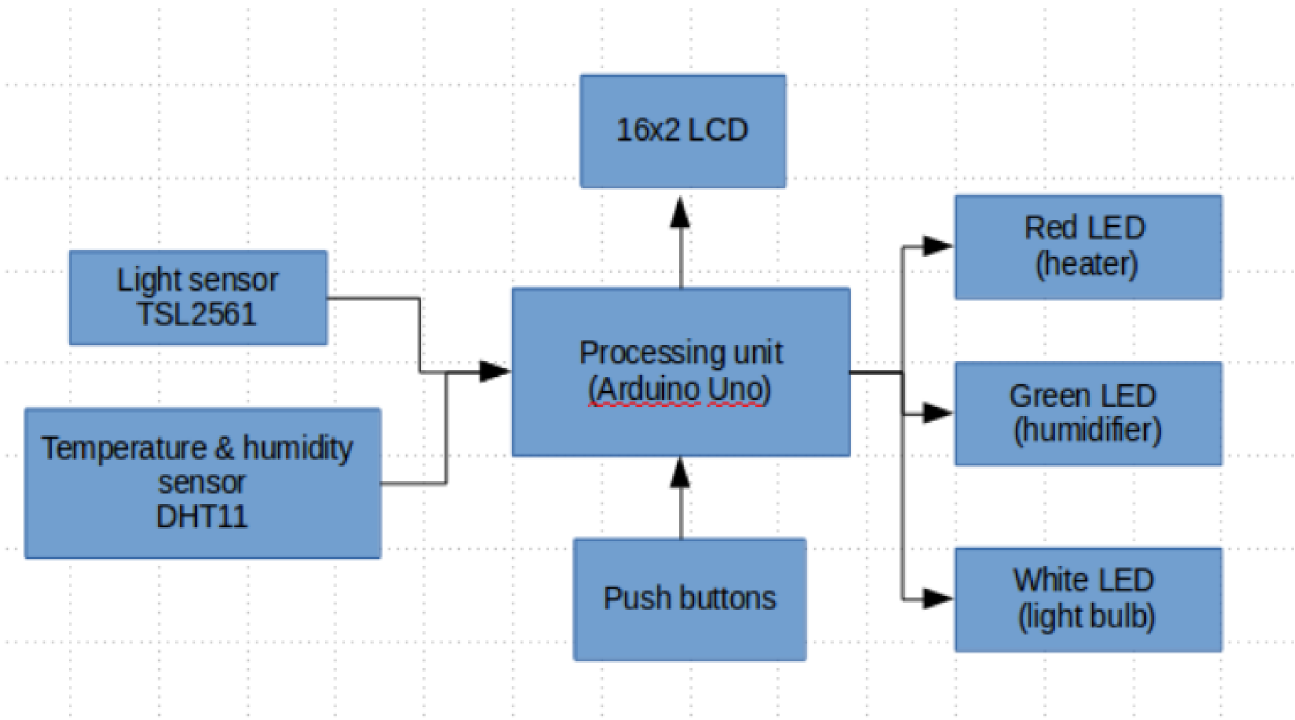

Figure 1. System block diagram of indoor environment control device

2. Equipment used

Type	Quantity
Arduino Uno	1
Light sensor TSL2561	1
Temperature and humidity sensor DHT11	1
16x2 LCD	1
Push buttons	2
Red LED	1
Green LED	1
White LED	1
USB type B – USB type A connector	1
Male to male wires	
Breadboard	1
10K potentiometer	1
10K resistor	5

3. Design of the device

3.1. Hardware connection

3.1.1. DHT11 temperature and humidity sensor

The sensor chosen for sensing humidity and temperature in this project is the DHT11 integrated sensor. This sensor is very basic and slow, but it meets the requirements for the project. It is made of two parts, a capacitive humidity sensor and a thermistor. An advantage of using DHT11 sensor is that it comes with an integrated chip which does some analogue to digital conversion and produces a digital signal with temperature and humidity. Digital signal is easy to read using the Arduino. Also, the DHT11 sensor is the off-the-shelf module which has been tested and ready to use, thus reducing the burden of designing electronic circuits which is time-consuming.

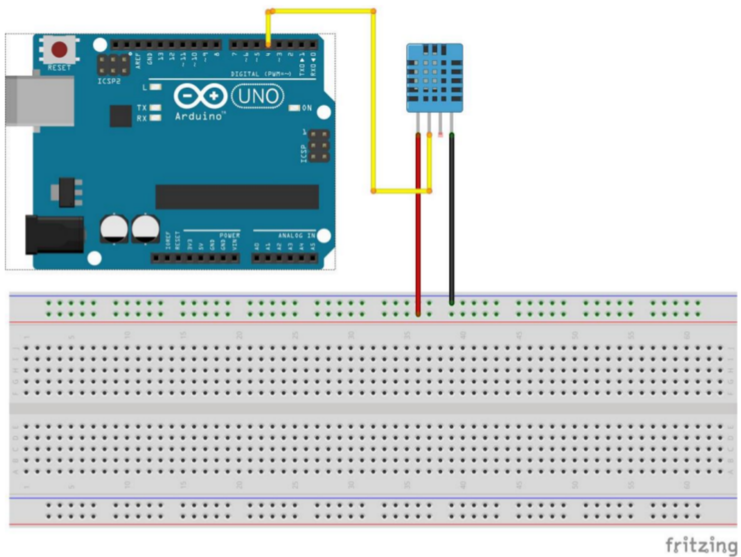

Figure 2. Connection between DHT11 and Arduino

The DHT11 sensor operates at 3 to 5 V, which is a good point since no external power converter is required. From 20 to 80% of humidity, the accuracy of humidity readings can be 5%. With temperature from 0 to 50 degrees, the accuracy of temperature readings is 2%. The sampling rate is quite slow, no more than 1Hz. However, this is enough for the project requirements. Using the DHT11 sensor saves the number of pins needed to connect to the Arduino Uno. In fact, it uses only one single digital pin of the Arduino Uno thanks to the single bus data communication. So, one single wire is used for communication and synchronization and is a two-way communication on a single wire. In the project, the DHT11 sensor will be connected to the Arduino Uno via Arduino digital pin 4. The 5V power supply for the sensor will be taken from the prototype board. Pin 4 of the sensor will be connected to the ground of the prototype board.

3.1.2. TSL2561 light sensor

To ease the development of the project, a digital light sensor has been chosen instead of an analogue sensor. The TSL2561 sensor comes with a breakout board for converting from 5v to 3.3V power supply for the sensor. The TSL2561 luminosity sensor is a digital light sensor. The best advantage of this sensor is that it contains both infrared and full spectrum diodes. That means it can be used to measure infrared, full-spectrum or human-visible light separately.

The sensor has an I2C interface. Three addresses can be selected so there can be up to three sensors on one board - each with a different i2c address. The built in ADC means that this sensor can be used to interface directly to the I2C pins of the ATmega328 microcontroller. The current draw is extremely low, about 0.5mA when actively sensing, and less than 15 uA when in power-down mode. The following gives brief information about the sensor's features:

Approximates human eye response
Temperature range: -30 to 80 degrees
Dynamic range (lux): 0.1 to 40.000 lux
Voltage range: 2.7 to 3.6 V
Interface: I2C

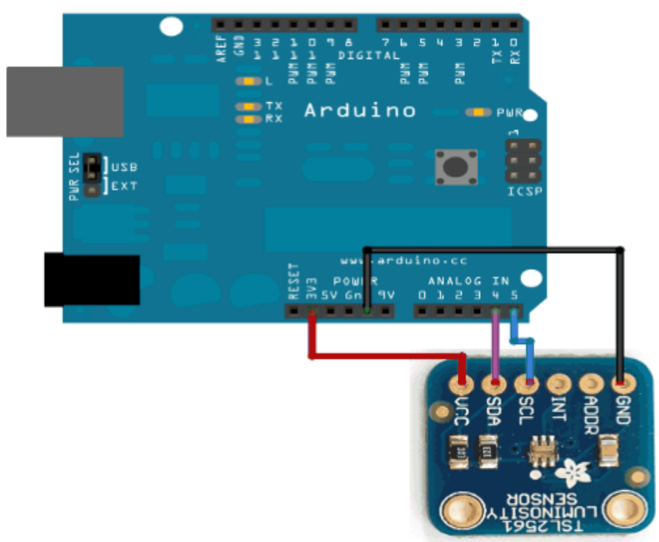

Figure 3. Connection between TSL2561 light sensor and Arduino

The I2C pins on the Arduino Uno platform is mapped to analogue pins 4 and 5. Analog pin 4 on Arduino Uno shall be connected to the data pin (SDA) of the sensor, and analog pin 5 on Arduino Uno shall be connected to the clock pin (SCL) of the sensor. The 5V power supply and the ground for the breakout board of the sensor are taken from the prototype board.

3.2. Design of user interface

The user interface consists of a 16x2 LCD, 2 push buttons with Schmitt trigger circuits for debouncing the buttons, and 3 red, green, and white LEDs. The following block diagram is used to better illustrate this user interface circuit. The following figure shows the user interface of the device.

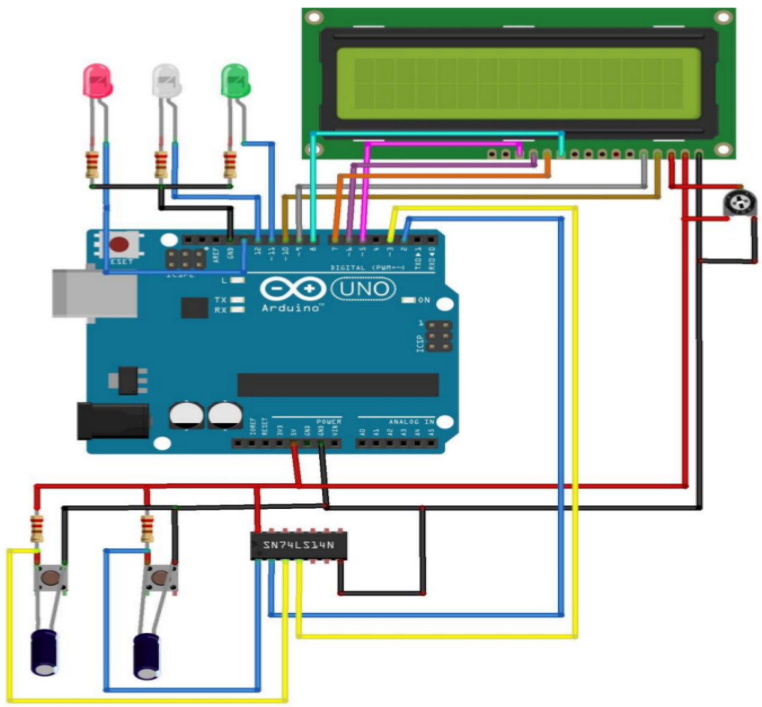

Figure 4. User interface of the device with push buttons and LCD

The 16x2 LCD is connected to the Arduino Uno according to the mapping as follows:

LCD pin name	LCD pin	Arduino Uno pin	Potentiometer pin
Vss	1	GND	3
Vcc	2	5V	1
Vo	3		2
Rs	4	10	
R/W	5	GND	
E	6	9	
D4	11	8	
D5	12	7	
D6	13	6	

D7	14	5

Pin 2 of the 10K potentiometer is connected to pin Vo of the LCD to adjust the contrast of the LCD. The button on the left-hand side in Figure 4 is connected to the Schmitt trigger circuit, which in turn is connected to Arduino Uno pin 2. The button on the right-hand side in Figure 4 is connected to also to the Schmitt trigger circuit, which in turn is connected to Arduino Uno pin 3. These buttons are used by a user to set the temperature and light intensity level and to turn on/off the light manually regardless of the light intensity level already set in the controlling unit. The green LED shall be connected to Arduino Uno pin 11. The white LED shall be connected to Arduino Uno pin 12, and the red LED shall be connected to Arduino pin 13 as seen in Figure 4.

3.3. Display of the information of the device

The16x2 LCD shall be used to display information about the device to the user. It also displays the interface for the user to set temperature and light intensity level as well as turn the light on/off regardless of the light intensity level. Specifically, it shall have five display modes, namely mode1, mode2, mode3, mode4, and mode5. The first mode (mode1) displays to the LCD information about the current temperature, humidity and light intensity level (in lux). All of these information must be displayed on the LCD at the same time. Mode2 displays the interface about temperature selection. In this mode, the user can set a desired temperature by pressing the right-hand side button. Mode3 displays the interface about humidity selection. In the same fashion as mode2, the user can set the humidity level by using the right-hand side button. Mode4 displays the interface about light intensity level selection. Like mode2 and mode3, this mode allows the user to select the light intensity level. For example, he/she can set the light intensity level to 450 lux which is the ideal level in an office. The last mode (mode 5) allows the user to turn light on/off manually regardless of the light intensity level read from the TSL2561 light sensor. The left-hand side button seen in Figure 4 shall be used by the user to select modes (e.g, mode1 or mode 2), while the right-hand side button shall be used to set temperature, light intensity, humidity, and turn the light on/off manually. Reading the status of the buttons should be implemented using interrupt since it is considered a priority. The following gives a rough interface of each mode that a user shall see on the 16x2 LCD.

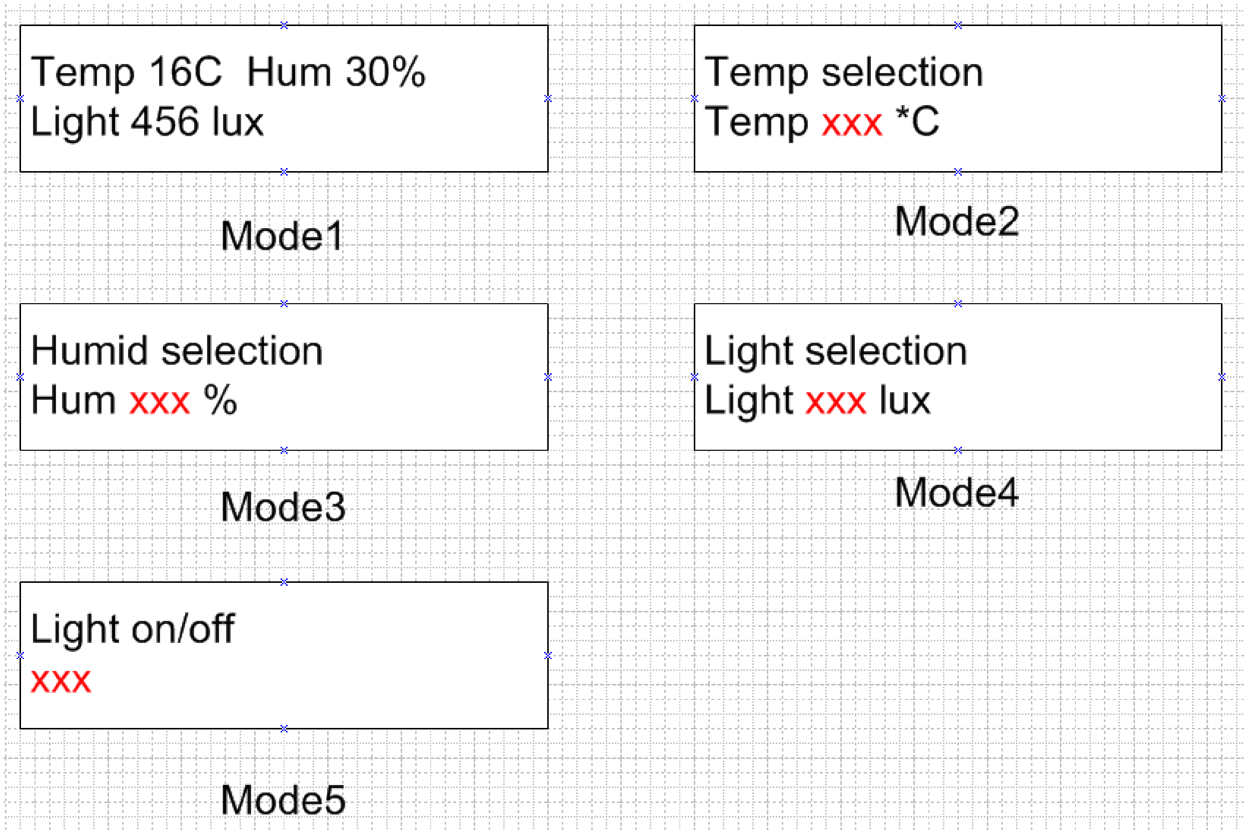

Figure 5. Selection modes displayed on LCD

The xxx in red is the variable that will be displayed on the LCD when the user selects temperature, light intensity level, humidity or light status (e.g, on or off).

3.4. Control of temperature, light intensity, and humidity

After power-on, the device will keep controlling temperature, light intensity and humidity. It compared the values read from the sensors with the value set by the user to turn on/off the LEDs accordingly. When the temperature, light intensity and humidity levels are higher than their threshold values set in the device, the controlling unit will turn off the corresponding LEDs. When these parameters are lower than their threshold values set in the device, the controlling unit will turn on the corresponding LEDs.

4. Implementation of the device

4.1. Hardware connection

Figure 6 shows the connection between hardware components of the device. All connection information is based on section 3 (Design of the device). The only thing which is different from the design is that Schmitt Trigger was not used since software can be used to solve button bouncing problem.

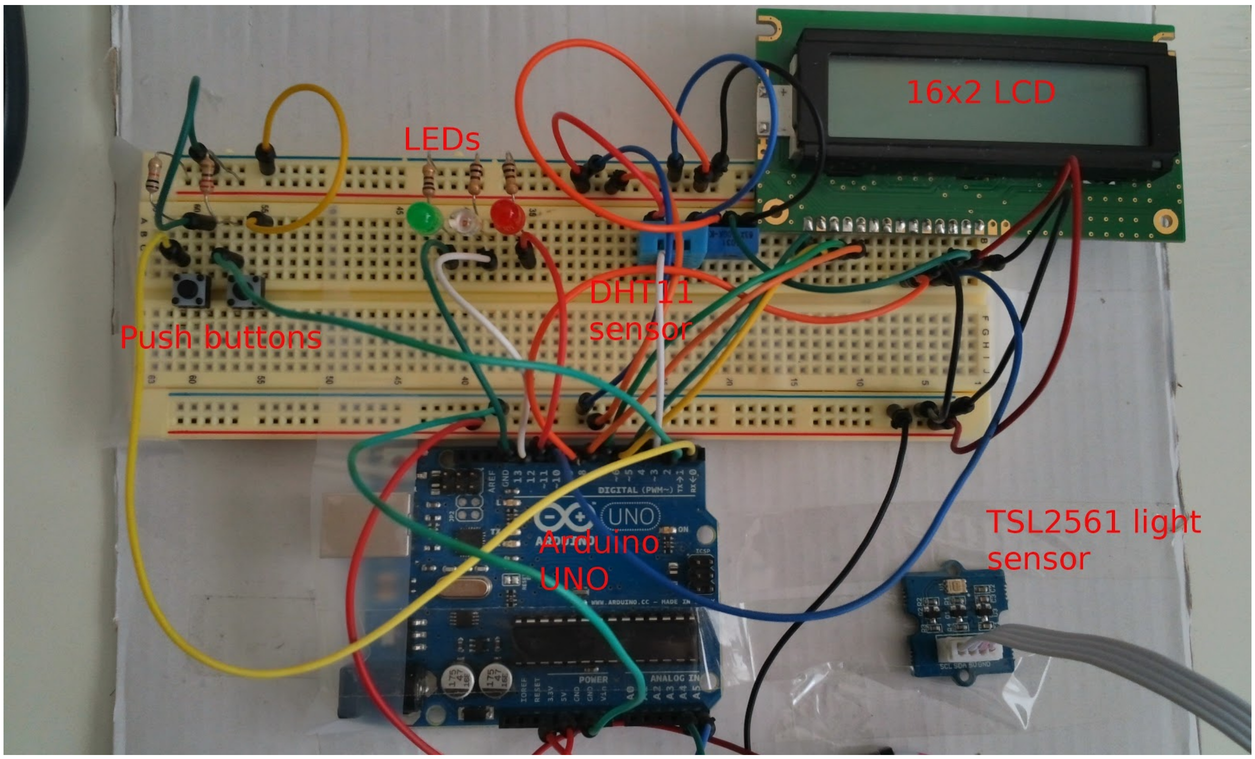

Figure 6. A prototype of the device

The Arduino UNO is used to give power supply to all the hardware components. The 3.3V power supply from Arduino Uno is used to power up the TSL2561 light sensor. The other 5V power supply of the Arduino UNO is connected to the power supply line of the breadboard which in turn provides voltage to the LEDs, the buttons for setting up pull-up configuration, to DHT11 sensor, and to the 16x2 LCD.

4.2. Embedded Software

4.2.1. Round-robin software architecture

The project does not have strict response-time requirements. Priority will be given when the device interacts with the user by means of interrupt. Therefore, a simple software architecture is used to ease the project software development work. In particular, round-robin with interrupts software architecture is used in the project. The code in Figure 7 is the prototype for round-robin with interrupts architecture. In this architecture, interrupt routines deal with the very urgent needs of the hardware and then set flags. The main loop polls the flags and does any follow-up processing required by the interrupts. This architecture gives more control over priorities when a user interacts with the system, setting temperature levels, for example.

```
Tboolean deviceAFlag = false;
Tboolean deviceBFlag = false;
…

Tboolean deviceZFlag = false;
void deviceA_ISR (void) {
       // Take care of I/O device A
       deviceAFlag = true;
}

void deviceB_ISR (void) {
       // Take care of I/O device B
       deviceBFlag = true;
}
…

void deviceZ_ISR (void) {
       // Take care of I/O device B
       deviceZFlag = true;
}

void main (void) {
       while (1) {
              if (deviceAFlag) {
                     deviceAFlag = false;
                     // Handle data to or from I/O device A
              }
              if (deviceBFlag) {
                     deviceBFlag = false;
                     // Handle data to or from I/O device B
              }
…

              if (deviceZFlag) {
                     deviceZFlag = false;
                     // Handle data to or from I/O device Z
              }
       } // End forever loop
} // End main() function
```

Figure 7. Round-robin with interrupt architecture

The interrupt routines can get good response because the hardware interrupt signal causes the microcontroller to stop what is doing in the task code and jump to serve the interrupt service routine instead.

4.2.2. Modular software design

The design of software is based on the modular software design architecture. Design of software is split into modules. Each of the modules can be a library or an application. The application is at the highest level of abstraction, and Arduino core library is at the lowest level of abstraction. The following figure shows the modular software design architecture in the project

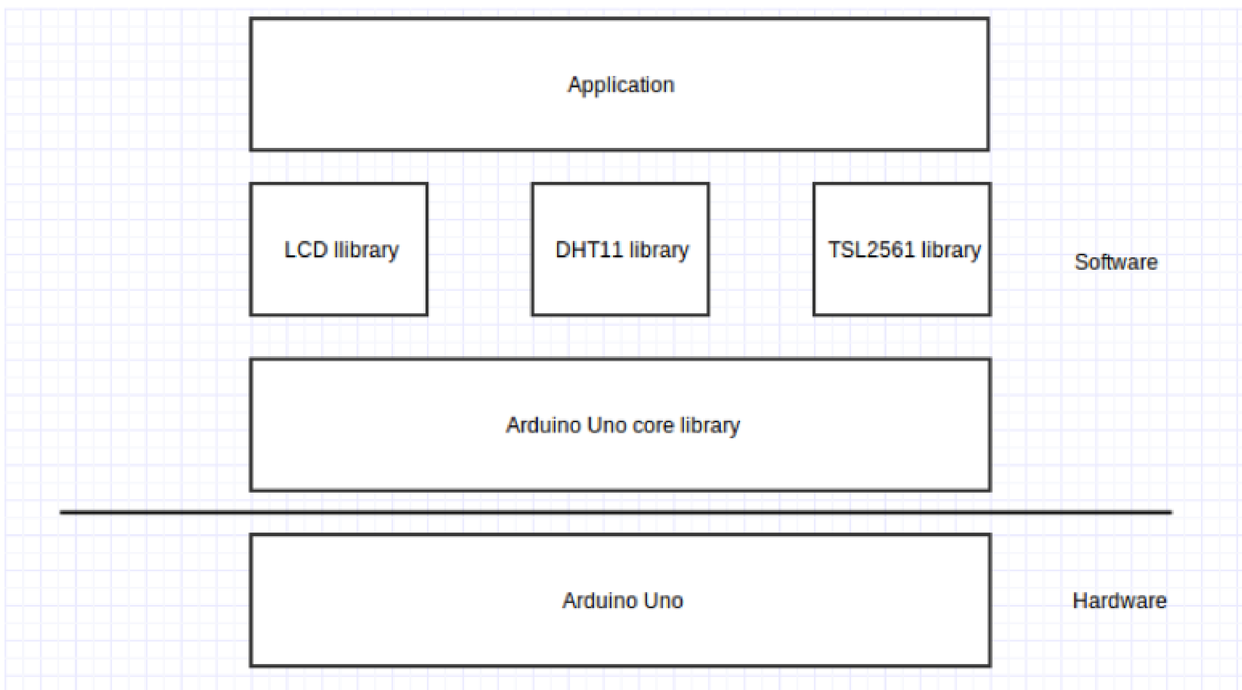

Figure 8. Modular software design architecture

The Arduino core library abstracts away the low level registers, memory mapping, interrupt mechanism and so on. It makes the software development a lot easier. The LCD library is already supported and has a higher level of abstraction than the Arduino core library. This library helps the application to display or clear text, set cursor, and so on when it needs to use the 16x2 LCD. The DHT11 and the TSL2561 libraries are at the same abstraction level as the LCD library. The DHT11 library will be written to support the application when it needs to get temperature and humidity level. It abstracts away the low level single bus communication between the Arduino Uno and the DHT11 sensor. The TSL2561 library functions in the same way as the DHT11 library in terms of abstraction. It hides detailed implementation of the I2C communication between the TSL2561 digital light sensor and the Arduino Uno and allows the application to get the

lux level of the surrounding environment. Finally, lying at the highest abstraction level is the application. The application is responsible for displaying information about the temperature, light and humidity levels to the 16x2 LCD, handling user interaction when he/she sets temperature, humidity and turns light on or off, and controlling three LEDs. The application and the libraries implemented in this project can be found from the following Github link.
https://github.com/quangng/Programming-Embedded-System-2015/tree/master/project

5. Testing of the device

5.1. Hardware testing

Since off-the-shelf modules such as digital light sensor, LCD, or temperature and humidity sensor are used in the project, they have been well tested by the manufacturers. The focus of hardware testing in the innovation project is to verify that the connection between these modules and the Arduino work as specified. The details of hardware testing plan is shown in the following table.

Table 1. Hardware testing test cases

No	Test case	Test purpose	Test description	Test result (pass/fail)
1	Arduino Uno platform test	This test is performed to verify that the Arduino Uno platform works properly after purchasing	A simple LED blinking application is written and programmed into the ATmega328 microcontroller. Observe that the LED on the Arduino Uno board blinks every second	Pass
2	LEDs test	This test is used for checking if the three red, white and green LEDs work. Also it verifies that the current limiting circuit works.	A simple LED blinking application is written and programmed into the ATmega328 microcontroller Observe that these three LEDs blink every second, and the light intensity from these LEDs can be seen by normal eyes	Pass

3	16x2 LCD test	This test is performed to verify that the 16x2 LCD module works and the connection between the LCD and the Arduino Uno is OK	Display the text “hello world” on both the first and the second line of the LCD. Check that “hello world” text is printed on the first and the second line of the LCD	Pass
4	DHT11 sensor test	This test case is executed to verify that the DHT11 sensor module works and that the connection between the DHT11 sensor and the Arduino Uno is OK	Use a ready-made library form DHT11 from the online community which has been tested to be working properly on Arduino Uno platform, and display the temperature and humidity level on the console. Compare the temperature reading from the console with a thermometer. Since a humidity tester is not available, testing of humidity reading of the sensor will be left out.	Pass

5	TSL2561 light sensor test	This test case is executed to verify that the TSL2561 sensor module works and that the connection between the TSL2561 sensor and the Arduino Uno is OK	Use a ready-made library form TSL2561 from the online community which has been tested to be working properly on Arduino Uno platform, and display the light level in lux on the console. A brand-new light bulb which can produce a lux level of 650 lux will be used to shine in a straight angle to the light sensor. Compare the light level reading on displayed on the console with light intensity that the light bulb can emit	Pass

Test cases were executed case by case, and it shows that all the components work fine.

5.2. Software testing

Since the project is based modular software design architecture, testing of software will be executed module by module. Each module will be tested using white-box testing method. The software in the project consists of 4 modules, namely DHT11 library, TSL2561 library, LCD library, and the application. The procedures for testing DHT11, TSL2561, and LCD libraries are the same as testing DHT11 sensor, TSL2561 sensor, and LCD display. Testing of the application requires more detailed test cases and is given in the following table. Testing of the application should follow the order in the following table.

Table 2. Software testing test cases

No	Test case	Test purpose	Test description	Test result (pass/fail)
1	User parameter setting	This test case will be performed to test that the device can respond fast enough when interacting with the user and that the selection of parameters by the user is within the limits	Change display mode on the LCD to temperature selection. Pressing the other pushbutton one at a time and observe that the temperature setting parameter displayed on the LCD is incremented from 20 to 30 and back to 20. Change display mode on the LCD to humidity selection. Pressing the other pushbutton one at a time and observe that the humidity setting parameter displayed on the LCD is incremented from 30 to 40 and back to 30 Change display mode on the LCD to light selection. Pressing the other pushbutton one at a time and observe that the light setting parameter displayed on the LCD displays ON and OFF	Pass
2	Humidity control test	This test case will be performed to test that the device can automatically control the humidity level set by the user	Set the humidity level to be 35%. Increase and keep the surrounding humidity to be over 35% and observe if the green led is turned off after 5 minutes Decrease and keep the surrounding humidity to be lower than 35% and observe if the green led is turned on immediately	Pass
3	Light control test	This test case is performed to test that the device can automatically control the light set by the user	Change to mode 5 and turn light on. Change to mode 4 and set the light level to be 450 lux. Shine the light bulb directly on the light sensor and observe if the white LED is turned after 1 minute. After that, use a finger	Pass

			to cover the light sensor and observe if the white LED is turned on immediately. Finally, go back to mode 5. Shine some light on the light sensor and use finger to cover the light sensor. Observe that the white LED should always be turned off.	
4	Temperature control test	This test case will be performed to test that the device can automatically control the temperature set by the user	Set the temperature to be 25 degrees. Increase and keep the surrounding temperature to be over 25 degrees and observe if the red led is turned off after 5 minutes. Decrease and keep the surrounding temperature to be lower than 25 degree and observe if the red led is turned on immediately	Pass

6. Conclusion

The aim of the project was to create a device capable of controlling temperature, humidity and light levels in a room automatically. Testing shows that the project was successful with all the test cases passed based on the test cases. However, there was a small problem with button bouncing. Sometimes, pressing a button once resulted in the counter incrementing two or three times. This suggests that a better push button should be used to deal with the problem for future project. The demonstration of the project can be seen from the following Youtube link
https://www.youtube.com/watch?v=Tn9DvczhTbE