

ELE8968 – Building Automation Systems

Review – Lab3 and Functions

Blerim Qela, Ph.D., P.Eng., SMIEEE

qelab@algonquincollege.com

Architecture, Civil, Electrical and Building Science

Copyright © B.Qela

Lab 3 – Temperature Sensing and Fan Control

Objective:

Implement a circuit which makes use of A/D converter and temperature sensor to read the current ambient temperature, and based on its threshold value - it controls the fan (ON/OFF). Combine the temperature sensor, A/D sensing capabilities with the actuation via a relay module to control an AC fan. Learn how to interface and control different devices. Utilize LCD and Serial Monitor for the display and/or debug purposes.

Equipment/Materials:

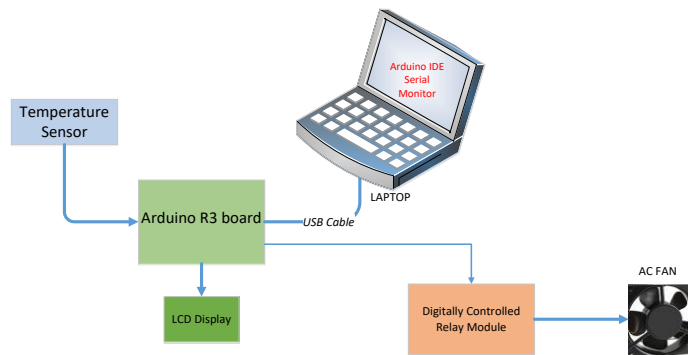
Tektronix Digital Oscilloscope	Digital Multimeter	Arduino Uno R3 Board
16x2 LCD Display	Arduino Software (IDE)	Grove Relay module
USB cable	10k potentiometer	Resistors/Capacitors (various)
Temperature sensor, LM35	AC Fan, Axial, 115 VAC	

ELE8968 – Building Automation Systems

Copyright © B.Qela

Lab 3 – Temperature Sensing and Fan Control

CAUTION!!!
DO NOT attempt this experiment without supervision of a qualified professional/instructor.



Block diagram of the temperature sensing and AC fan control

ELE8968 – Building Automation Systems
Copyright © B.Qela

3

Lab 3 – Temperature Sensing and Fan Control

Use the Arduino tools and components listed above to build the circuit. The circuit updates constantly the temperature reading on LCD Display. Furthermore, it indicates if the AC Fan is ON or OFF.

The **Fan_ ON** procedure should be implemented as a **function** and invoked via function call method from the main. The fan should turn ON for any temperature reading above the actual **Room Temperature**, and it should turn OFF as the temperature sensor reading falls below or is equal to the **Room Temperature**. This should be acknowledged by turning ON an LED indicator.

ELE8968 – Building Automation Systems
Copyright © B.Qela

4

Lab 3 – Temperature Sensing and Fan Control

Another important detail to implement in your laboratory experiment is the following:

- a) After the *first time* that temperature reading is above the actual **Room Temperature**, and the **Fan_ON** procedure is triggered, your program should ensure to maintain the Fan ON for an extra 15 seconds after the temperature drops at **Room Temperature** (or below). After which, Fan should turn OFF.
- b) After the *second time* that temperature reading is above the actual **Room Temperature**, and the **Fan_ON** procedure is triggered, your program should ensure to maintain the Fan ON for another extra 10 seconds after the temperature drops at **Room Temperature** or below. After which, Fan should turn OFF.

*ELE8968 – Building Automation Systems
Copyright © B.Qela*

5

Lab 3 – Temperature Sensing and Fan Control

An important detail to implement in your laboratory experiment is the following:

- c) After the *third time* that temperature reading is above the actual **Room Temperature**, and the **Fan_ON** procedure is triggered, your program should ensure to maintain the Fan ON for another extra 5 seconds after the temperature drops at **Room Temperature** (or below). After which, Fan should turn OFF.

Note: *In each of the above cases (a, b or c) LCD should display the case that is running (e.g., **Case a running**) to ensure that the demo is evaluated and marked correctly. The **Fan_ON** procedure should be implemented as a function and invoked via function call from the main.*

*ELE8968 – Building Automation Systems
Copyright © B.Qela*

6

Functions – Review and Examples

- What are functions?
 - Subprograms
- How important are subprograms in modern programming languages?
 - Subprograms are essential part - core of the modern programming languages!
 - Why?
 - They provide the means to enlarge the set of elementary building blocks with which we program
 - They enable abstraction, re-use and provide the means not to repeat the code in the same program!

ELE8968 – Building Automation Systems
 Copyright © B.Qela

6

Functions – Review and Examples

- Functions break large computing tasks into smaller ones
- Programming language is essentially a collection of functions
- Functions simply contain a group of statements/expressions which are executed whenever the function is called

ELE8968 – Building Automation Systems
 Copyright © B.Qela

7

Functions – Review and Examples

- How can we implement a function?
 1. Function needs to be declared (**Function Prototype**)
 2. Function needs to be defined (**Function Definition**)
- What is the purpose of function declaration (**Function Prototype**)?
 - To inform the compiler about relevant properties of the function...How?
 - In the similar way that variable's declaration informs the compiler about the variable being declared
 - e.g. int x (compiler is informed about the declaration of the variable x of type integer)

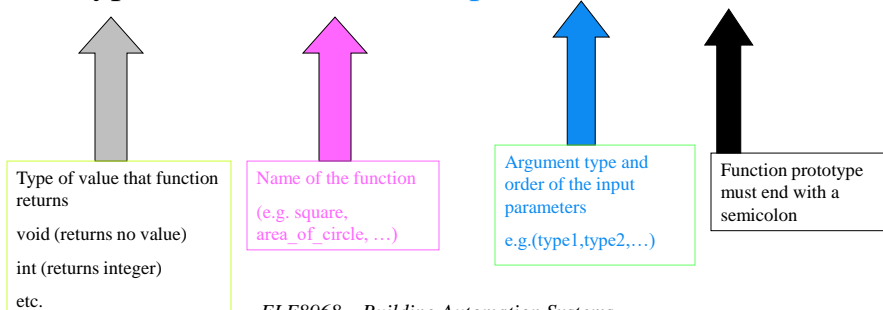
ELE8968 – Building Automation Systems
Copyright © B.Qela

8

Functions – Review and Examples (cont'd)

- How are the functions declared (**Function Prototype**) in C or C-like languages?

type **name_of_function** (**parameter list**) ;



ELE8968 – Building Automation Systems
Copyright © B.Qela

9

Functions – Review and Examples

- What is the purpose of **Function Definition**?
 - To define the action to be performed by a function declared in a function prototype
 - Remember that having meaningful name for the functions is important!
 - To perform a specific task (print to the screen (message, info, ...), calculate the area of a circle, generate a random number, calculate square root of a number, etc.)

ELE8968 – Building Automation Systems
 Copyright © B.Qela

10

Functions – Review and Examples

- How are the functions defined (**Function Definition**) in C or C-like languages?

```
type name_of_function (parameter list )
{
  declarations ;
  statements (commands, expressions);
}
```

ELE8968 – Building Automation Systems
 Copyright © B.Qela

11

Functions – Review and Examples

- Two ways of calling a function and passing the arguments to a function:
 - Call by value
 - A copy of the argument is passed to the function; thus changes made to a copy of the argument do not affect the argument
 - Call by reference
 - the address of the argument itself is copied into a parameter → passed to the function; thus changes made to the parameter affect the argument

ELE8968 – Building Automation Systems
Copyright © B.Qela

12

Functions – Review and Examples

```
void setup(){
  Serial.begin(9600);
}

void loop() {
  int a = 1;
  int b = 2;
  int c;

  c = addFunction(a, b); //function call
  Serial.print("Result is: ");
  Serial.print(c); // print the result of addition
  Serial.print('\n'); // newline character
  delay(1000);
}

//This is a simple function which adds two numbers
int addFunction(int a, int b)
{
  int result;
  result = a + b;
  return result;
}
```

*Arduino - Sample
function implementation.
Is this call by value or
call by reference?*

ELE8968 – Building Automation Systems
Copyright © B.Qela

14