

Laboratory 8

Title of the Laboratory Exercise: Introduction to Arduino UNO

1. Introduction and Purpose of Experiment

Students will be able to perform basic programming on Arduino UNO board

2. Aim and Objectives

Aim

To understand Arduino programming language and to develop basic programs using Arduino programming language.

Objectives

At the end of this lab, the student will be able to

- Explain analog and digital pins in Arduino
- Basic hardware programming language
- Interface sensors and read values from sensors
- Drive actuators

3. Experimental Procedure

1. Write algorithm to solve the given problem
2. Translate the algorithm to Arduino programming language
3. Execute it in Arduino IDE
4. Create a laboratory report documenting the work

4. Questions

Perform the following:

1. Print hello world in Arduino
 2. Set High and LOW value to digital pin and print its output in serial monitor
 3. Read Analog values and print it in serial monitor
- ### 5. Calculations/Computations/Algorithms

```
sketch_oct30a $  
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600);  
  pinMode(13,OUTPUT);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  int value;  
  Serial.println("\nHello");  
  digitalWrite(13,HIGH);  
  delay(1000);  
  value=digitalRead(13);  
  Serial.println(value);  
  digitalWrite(13,LOW);  
  delay(1000);  
  value=digitalRead(13);  
  Serial.println(value);  
}  
  
Done compiling.  
  
Sketch uses 2404 bytes (7%) of program storage space. Maximum is 32256 bytes.  
Global variables use 194 bytes (9%) of dynamic memory, leaving 1854 bytes for local variables. Maximum is 2048 bytes.
```

```
sketch_oct30a  
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  Serial.println("Hi");  
}//  
  
Done compiling.  
  
Sketch uses 1476 bytes (4%) of program storage space. Maximum is 32256 bytes.  
Global variables use 190 bytes (9%) of dynamic memory, leaving 1858 bytes for local variables. Maximum is 2048 bytes.
```

6. Analysis and Discussions:

Examples of basic Arduino programs:-

The structure of Arduino program is pretty simple. Arduino programs have a minimum of 2 blocks,

Preparation & Execution

Each block has a set of statements enclosed in curly braces:

```
void setup( )  
  
{  
  
statements-1;  
  
.  
  
.  
  
.  
  
statement-n;  
  
}  
  
void loop ( )  
  
{  
  
statement-1;  
  
.  
  
.  
  
.  
  
statement-n;  
  
}
```

Here, setup () is the preparation block and loop () is an execution block.

The setup function is the first to execute when the program is executed, and this function is called only once. The setup function is used to initialize the pin modes and start serial communication. This function has to be included even if there are no statements to execute.

```
void setup ( )
```

```
{
```

```
pinMode (pin-number, OUTPUT); // set the 'pin-number' as output
```

```
pinMode (pin-number, INPUT); // set the 'pin-number' as output
```

```
}
```

After the setup () function is executed, the execution block runs next. The execution block hosts statements like reading inputs, triggering outputs, checking conditions etc..

In the above example loop () function is a part of execution block. As the name suggests, the loop () function executes the set of statements (enclosed in curly braces) repeatedly.

```
Void loop ( )
```

```
{
```

```
digitalWrite (pin-number,HIGH); // turns ON the component connected to 'pin-number'
```

```
delay (1000); // wait for 1 sec
```

```
digitalWrite (pin-number, LOW); // turns OFF the component connected to 'pin-number'
```

```
delay (1000); //wait for 1sec
```

```
}
```

Note: Arduino always measures the time duration in millisecond. Therefore, whenever you mention the delay, keep it in milli seconds.

7. Conclusions :

Basic programs are successfully developed in Arduino programming language.

Signature and date

Marks

