NAME – ISHAN DATE – 16/08/2024

ROLL NO. – FET-BAML-2022-26-012

SUBJECT – IOT AND AUTOMATION

**ASSINGMENT-3**

**1.Explain Basic Structure of an Arduino Program**

**ANS-** An Arduino program (or sketch) typically consists of two main functions: setup() and loop(). These functions define the basic structure of the program, enabling the microcontroller to initialize settings and execute repeated actions.

Here’s a breakdown of the basic structure of an Arduino program:

1. Preprocessor Directives

* These include any #include statements or #define statements, used to include libraries or define constants

Code example-

#include <Servo.h> // Example of including a library

#define LED\_PIN 13 // Define a constant for the LED pin

2. Global Variable Declarations

* Global variables are declared outside of any function. These variables can be used by both setup() and loop().

Code example-

int ledPin = 13; // Declare a global variable for the LED pin

3. setup() Function

* The setup() function runs once when the microcontroller starts or when it is reset. It is typically used to configure pin modes, initialize libraries, and set up communication.
* Example uses: setting pin modes, starting serial communication, initializing sensors, etc.

Code example-

void setup() {

pinMode(ledPin, OUTPUT); // Set the LED pin as an output

Serial.begin(9600); // Start serial communication at 9600 baud rate

}

4. loop() Function

* The loop() function contains the main logic of the program. After setup() completes, loop() runs repeatedly in a cycle. This is where the continuous execution of tasks, such as reading sensors, controlling outputs, and making decisions, happens.
* Example: turning an LED on and off, reading sensor data, etc.

Code example-

void loop() {

digitalWrite(ledPin, HIGH); // Turn the LED on

delay(1000); // Wait for 1 second

digitalWrite(ledPin, LOW); // Turn the LED off

delay(1000); // Wait for 1 second

6. Comments

* Comments in the code are used to explain what different parts of the program do. These are ignored by the compiler but are essential for making the code understandable.
* Single-line comments start with //.
* Multi-line comments are enclosed within /\* \*/.

Code example-

// This is a single-line comment

/\*

This is a

multi-line comment

\*/

Example of a Basic Arduino Sketch

Code example-

#include <Servo.h> // Example of a library inclusion

#define LED\_PIN 13 // Define constant for LED pin

int buttonPin = 7; // Declare a global variable for the button pin

void setup() {

pinMode(LED\_PIN, OUTPUT); // Set LED pin as output

pinMode(buttonPin, INPUT); // Set button pin as input

Serial.begin(9600); // Initialize serial communication

}

void loop() {

int buttonState = digitalRead(buttonPin); // Read the button state

if (buttonState == HIGH) {

digitalWrite(LED\_PIN, HIGH); // Turn LED on if button is pressed

Serial.println("Button Pressed"); // Print message to serial monitor

} else {

digitalWrite(LED\_PIN, LOW); // Turn LED off if button is not pressed

}

delay(100); // Short delay to debounce the button

}

**Key Points:**

* setup() runs once and is used for initialization.
* loop() runs repeatedly and contains the main logic of the program.
* Global variables and custom functions help organize and reuse code.
* Preprocessor directives help manage external libraries and constants.