Introduction to C Programming for Arduino

A VERY BASIC OVERVIEW

Overview

- Basic C syntax rules
- Variables
- Operators
- Conditional statements
- Loops
- Functions

Basic C Syntax Rules

- C is case-sensitive
- Every statement ends with ;
- Blocks of code use { }
- Comments:

```
// Single line comment
/* Multi-line
   comment */
```

Variables (Storing Data)

Variables store values in memory.

```
int ledPin = 13;
//int → integer number
//ledPin → variable name
//13 → value stored
```

- Common variable types:
  - int → whole numbers
  - float → decimal numbers
  - char → single character
  - bool → true / false

Using Variables in Arduino

```
int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
}
```

- Variables make code:
  - Easier to read
  - Easier to modify
  - Change pin number in one place

Operators (Basic Math & Logic)

- Arithmetic operators:
  - + addition
  - - subtraction
  - * multiplication
  - / division
- Comparison operators:
  - == equal to
  - != not equal
  - <, >, <=, >=

Conditional Statements (if–else)

Used to make decisions

```
if (value > 500) {
  digitalWrite(13, HIGH);
} else {
  digitalWrite(13, LOW);
}
```

- Arduino reacts based on conditions
- Essential for sensors

Loops (Repeating Actions) : ‘for’ loop

Runs code fixed number of times

```
for (int i = 0; i < 5; i++) {
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
}
```

Loops (Repeating Actions) : ‘while’ loop

- Runs as long as condition is true
- Useful for waiting for events

```
while (buttonState == LOW) {
  digitalWrite(13, HIGH);
}
```

Functions (Reusable Code)

Functions help organize code.

```
void blinkLED() {
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
}
```

Calling a function

```
blinkLED();
```

Serial Communication (Debugging)

```
Serial.begin(9600);
Serial.println(sensorValue);
```

- Used to:
  - Print values
  - Debug programs
  - View output in Serial Monitor

Thank You

Any Questions