NanoBoy User Manual

The **NanoBoy** is a fun and educational embedded systems board, developed by **Mikrotronics Pakistan**. It is designed for students and hobbyists to learn embedded programming in a playful way, using familiar concepts such as *games, sprites, tiles, and simple UI interfaces*. Unlike traditional microcontroller projects, NanoBoy abstracts away low-level hardware details and instead provides a simple, high-level **Arduino-based library** for working with graphics, input buttons, sound, and LEDs. This approach makes NanoBoy both a **learning platform** and a **game development kit**, allowing you to explore C/C++ programming while creating interactive projects.

Library Overview

The NanoBoy library abstracts the OLED display, buttons, buzzer, and LED. It exposes functions to draw graphics, manage text, create and move sprites, detect collisions, and organize tile maps. This section explains each function in the library with: Syntax Parameters Description Example code

Text Functions

Syntax: void setCursor(int x, int y)
Description: Sets the position where text will appear on the screen.

```
nb.setCursor(0, 0);
nb.print("Hello, NanoBoy!");
```

Syntax: void print(const char* text)
Description: Prints text at the current cursor position.

```
nb.setCursor(10, 20);
nb.print("Score: 10");
```

Syntax: void print(int num) / void print(float num)
Description: Prints numeric values (integer or floating-point).

```
nb.setCursor(10, 40);
nb.print(123);
nb.print(3.14);
```

Graphics Functions

Syntax: void drawPixel(int x, int y)
Description: Draws a single pixel at (x,y).

```
nb.drawPixel(64, 32);
```

Syntax: void drawRect(int x, int y, int w, int h)
Description: Draws a rectangle outline.

```
nb.drawRect(20, 20, 40, 20);
```

Syntax: void fillRect(int x, int y, int w, int h)
Description: Draws a filled rectangle.

```
nb.fillRect(10, 10, 8, 8);
```

Syntax: void drawCircle(int x, int y, int r)
Description: Draws a circle outline.

```
nb.drawCircle(64, 32, 10);
```

Syntax: void drawLine(int x0, int y0, int x1, int y1)
Description: Draws a straight line.

```
nb.drawLine(0, 0, 127, 63);
```

Sprites and Collision

Syntax: struct Sprite { int x, y, w, h; const uint8_t* bitmap; bool active; };
Description: A Sprite represents a movable game object with a position, size, and optional bitmap.

```
Sprite player = {10, 20, 8, 8, playerBitmap, true};
nb.drawSprite(player);
```

Syntax: void drawSprite(const Sprite& s)
Description: Draws a sprite on screen.

```
nb.drawSprite(player);
```

Syntax: bool checkCollision(const Sprite& a, const Sprite& b)
Description: Detects collision between two sprites.

```
if (nb.checkCollision(player, enemy)) { nb.print("Hit!"); }
```

Input & Sound

Syntax: bool buttonPressed(int btn)
Description: Checks if a button is pressed.

```
if (nb.buttonPressed(BTN_A)) nb.print("Pressed A");
```

Syntax: void beep(int freq, int dur)
Description: Plays a beep sound at frequency (Hz) for duration (ms).

```
nb.beep(440, 200);
```

Example Games

```
<b>Example 1: Pong (simplified)</b>

#include <NanoBoy.h>
NanoBoy nb;

int ballX = 64, ballY = 32, dx = 1, dy = 1;

void setup() {
  nb.begin();
}

void loop() {
  nb.clear();
  nb.fillRect(ballX, ballY, 4, 4);   // Draw ball
  ballX += dx; ballY += dy;

  if (ballX <= 0 || ballX >= 124) dx = -dx;
  if (ballY <= 0 || ballY >= 60) dy = -dy;

  nb.display();
  delay(50);
}
<b>Example 2: Text Demo</b>

#include <NanoBoy.h>
NanoBoy nb;

void setup() {
  nb.begin();
}

void loop() {
  nb.clear();
  nb.setCursor(10, 20);
  nb.print("Hello World");
  nb.setCursor(10, 40);
  nb.print(1234);
  nb.display();
  delay(1000);
}
```