ab9nq.william@gmail.com

Could write operation of the switches in the following Arduino C++ code?

Certainly! Here is an explanation of the switch statements in the following Arduino C++ code:

c++

Copy code

```
void loop() {
  switch (digitalRead(buttonPin)) {
    case HIGH:
      digitalWrite(ledPin, HIGH);
      break;
    case LOW:
      digitalWrite(ledPin, LOW);
      break;
  }
}
```

This code is meant to read the state of a button connected to an input pin (specified as buttonPin), and then turn on or off an LED connected to an output pin (specified as ledPin) based on the button state.

The switch statement is used to check the state of the button. It takes the value returned by digitalRead(buttonPin) as its argument, which will either be HIGH or LOW depending on whether the button is currently pressed or not.

If the button is pressed (i.e., if digitalRead(buttonPin) returns HIGH), then the case HIGH statement is executed. This statement sets the state of the LED to HIGH, turning it on.

If the button is not pressed (i.e., if digitalRead(buttonPin) returns LOW), then the case LOW statement is executed. This statement sets the state of the LED to LOW, turning it off.

Note that

ab9nq.william@gmail.com

Forgot to include the code:

```
void loop() {

  uint32_t previous = millis();

  //  if ((millis()-previous) > 500) {
  //    previous = millis();
  //          //do something
  //  }

  for (int x = 1; x < 5000; x++) {
    ftpSrv.handleFTP();
  }

  if(toogleSWITCH == 1){
    Serial.println("Interrupt occured");
    toogleSWITCH = 0;
  }

  // Check if switch 1 has been toggled
  if (switch1State != switch1PrevState) {
    // If the switch is now pressed
    if (switch1State == LOW) {
      Serial.println("Switch 1a pressed");
  // Toggle the LED
  ledState = !ledState;
  digitalWrite(ledPin, ledState);
  // Disable switch 2 if LED is off
  if (ledState == LOW){
   switch2State = LOW;  // Switch 2 dusabled
   Serial.println("Going to Deep Sleep");
   goToDeepSleep();
  }
 }
 // Update the previous state of the switch
 switch1PrevState = switch1State;
}

// Check if switch 1 has been toggled
if (switch1State != switch1PrevState) {
 // If the switch is now pressed
 if (switch1State == HIGH) {
  Serial.println("Switch 1b pressed");
  // Toggle the LED
  ledState = !ledState;
  digitalWrite(ledPin, ledState);
  // Disable switch 2 if LED is off
  if (ledState == HIGH) {
  // Switch 2 enabled
   Serial.println("Live");
  }
 }
 // Update the previous state of the switch
```

```
  switch1PrevState = switch1State;
}

// Check if switch 2 has been toggled
if (ledState == HIGH && switch2State != switch2PrevState) {
  // If the switch is now pressed
  if (switch2State == HIGH) {
    Serial.println("Switch 2a pressed");
    // Call function 1 or function 2 depending on the current state of the LED
    if (ledState == HIGH) {
      isRGBRunning = true;
      for (int x = 1; x < 2; x++) {
        Serial.println("Motion");
      }
      motion();
    }
  }
  // Update the previous state of the switch
  switch2PrevState = switch2State;
}

// Check if switch 2 has been toggled
if (ledState == HIGH && switch2State == switch2PrevState) {
  // If the switch is now pressed
  if (switch2State == LOW) {
    Serial.println("Switch 2b pressed");
    // Call function 1 or function 2 depending on the current state of the LED
    if (ledState == HIGH) {
      for (int x = 1; x < 2; x++) {
```

```
     Serial.println("noMotion");
    }
    isRGBRunning = true;
    noMotion();
   }
  }
  // Update the previous state of the switch
  switch2PrevState = switch2State;
 }

 // Debounce switch 1
 if ((millis() - switch1LastDebounceTime) > debounceDelay) {
  switch1State = digitalRead(switch1Pin);
  switch1LastDebounceTime = millis();
 }

 // Debounce switch 2
 if ((millis() - switch2LastDebounceTime) > debounceDelay) {
  switch2State = digitalRead(switch2Pin);
  switch2LastDebounceTime = millis();
 }
}
```

Certainly! Here is an explanation of the switch statements in the following Arduino C++ code:

c++

Copy code

```
// Check if switch 1 has been toggled
if (switch1State != switch1PrevState) {
 // If the switch is now pressed
```

```
  if (switch1State == LOW) {
    Serial.println("Switch 1a pressed");
    // Toggle the LED
    ledState = !ledState;
    digitalWrite(ledPin, ledState);
    // Disable switch 2 if LED is off
    if (ledState == LOW){
      switch2State = LOW;  // Switch 2 dusabled
      Serial.println("Going to Deep Sleep");
      goToDeepSleep();
    }
  }
  // Update the previous state of the switch
  switch1PrevState = switch1State;
}

// Check if switch 1 has been toggled
if (switch1State != switch1PrevState) {
  // If the switch is now pressed
  if (switch1State == HIGH) {
    Serial.println("Switch 1b pressed");
    // Toggle the LED
    ledState = !ledState;
    digitalWrite(ledPin, ledState);
    // Disable switch 2 if LED is off
    if (ledState == HIGH) {
    // Switch 2 enabled
      Serial.println("Live");
    }
  }
  // Update the previous state of the switch
  switch1PrevState = switch1State;
}

// Check if switch 2 has been toggled
if (ledState == HIGH && switch2State != switch2PrevState) {
  // If the switch is now pressed
  if (switch2State == HIGH) {
    Serial.println("Switch 2a pressed");
    // Call function 1 or function 2 depending on the current state of the LED
    if (ledState == HIGH) {
      isRGBRunning = true;
      for (int x = 1; x < 2; x++) {
        Serial.println("Motion");
      }
      motion();
    }
  }
  // Update the previous state of the switch
  switch2PrevState = switch2State;
}

// Check if switch 2 has been toggled
if (ledState == HIGH && switch2State == switch2PrevState) {
  // If the switch is now pressed
  if (switch2State == LOW) {
    Serial.println("Switch 2b pressed");
    // Call function 1 or function 2 depending on the current state of the LED
```

```
   if (ledState == HIGH) {
     for (int x = 1; x < 2; x++) {
       Serial.println("noMotion");
     }
     isRGBRunning = true;
     noMotion();
   }
  }
  // Update the previous state of the switch
  switch2PrevState = switch2State;
}
```

These switch statements are used to check the state of two switches connected to input pins (switch1Pin and switch2Pin) and then execute different blocks of code depending on the state of the switches.

The first switch statement checks if switch1State is different from switch1PrevState. If it is, the code inside the switch statement is executed. If switch1State is LOW, the LED state is toggled, and if the LED state is LOW, switch2State is set to LOW, which disables switch2. If switch1State is HIGH, the LED state is toggled and switch2 is enabled. In either case, the previous state of switch1State is updated.

The second switch statement is nearly identical to the first

ab9nq.william@gmail.com

How would a user use switches in this sketch form no power to sellecting effects to deep sleep to awaken from deep sleep?

In this sketch, there are two switches, switch 1 and switch 2, which are connected to Arduino digital pins. The user can interact with these switches to control the behavior of the sketch. Here is a step-by-step guide to using the switches to perform various functions:

Power on the Arduino board: Connect the board to a power source to turn it on.

Initial state: When the sketch starts running, the LED connected to the board will be off, and switch 2 will be disabled.

Select an effect: Press switch 1a to turn on the LED and select an effect. When the LED is turned on, switch 2 will be disabled. If you press switch 1b, the LED will be turned off, and switch 2 will be enabled.

Activate an effect: Press switch 2a to activate the effect selected in step 3. This will trigger the "motion" function, which will turn on a motor or trigger a motion sensor, depending on the hardware connected to the board. If you press switch 2b, the "noMotion" function will be activated.

Deep sleep: If the LED is turned off (step 3b), the board will enter deep sleep mode automatically. The sketch will not execute any code until it is awakened from deep sleep.

Wake up from deep sleep: To wake up the board from deep sleep, you can press switch 1a or 1b. The board will then continue executing the sketch from the point where it left off before entering deep sleep.

Note that there is a debounce function implemented to prevent unintended activation of the switches due to bouncing. The debounceDelay variable determines the time in milliseconds that the sketch will wait after detecting a switch state change before reading the switch state again.

ab9nq.william@gmail.com

There are two more missing "items" third switch SW 3 Interrupt infinite loop of moMotion and motion functions. SW 2 also serves to awaken from deep sleep, requires pressing SW 1 to toogle built-in LED.

Thank you for the clarification. Here's an updated description of how a user could use the switches in the sketch:

Power On: Connect the power source to the Arduino board to power it on.

Select Effects: Press switch SW1 to toggle the built-in LED on and off. While the LED is on, press switch SW2 to select between the "motion" and "noMotion" effects. When switch SW2 is pressed, the corresponding effect will be triggered based on the current state of the LED. If the LED is off, switch SW2 will be disabled.

Deep Sleep: If the LED is off, press switch SW1 to toggle it on. This will disable switch SW2. When the LED is on, press switch SW1 again to initiate a deep sleep mode. The board will enter a low-power state and remain in this state until it is awakened by pressing switch SW2.

Wake Up: Press switch SW2 to awaken the board from deep sleep. If the LED is off, pressing switch SW2 will have no effect. Once the board has been awakened, the LED will be toggled on and off again using switch SW1, and the effects can be selected using switch SW2 as described in step 2.