**Lab Report # 02**

**Name**\_\_\_\_\_\_\_Ruvail Shahzad\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Introduction:**

I had some experience with Arduino programming but this lab helped me understand how, implementation of similar microprocessors, with I/O options increase the possibilities and ease the accessibility for any embedded system.

**Procedure:**

I programmed the WIO terminal to perform a specific task, there are basically 2 main part of the code.

* Setup()

In setup I initialized my variables and setup the modes for the pins, (buttons, led and buzzer).

* Loop()

The loop function in code is a sentinel loop which will keep running until the device is turned off, I have used ”If”, “elseif” blocks to check each button one at a time and if any switch is pressed (connected to ground) ie. **digitalRead(WIO\_KEY\_B) == LOW** then we perform the required task. I have explained in next section.

**Explain Code Snippets:**

* ***Functions Prototyping:***

|  |
| --- |
| #define BUZZER\_PIN WIO\_BUZZER  //function prototypes (Arduino make them automatically in most cases)  int **seriesB**(int);  int **seriesA**(int);  void **blinking**(int);  void **buzzer**(); |

Wio terminal can do the prototyping itself, but its good practice to do it manually.

* **Setup:**

|  |
| --- |
| void setup() {  int ButtonA\_press\_count=0;  int ButtonB\_press\_count=0;  Serial.begin(115200);  pinMode(WIO\_KEY\_A, INPUT\_PULLUP);  pinMode(WIO\_KEY\_B, INPUT\_PULLUP);  pinMode(WIO\_KEY\_C, INPUT\_PULLUP);  pinMode(WIO\_BUZZER, OUTPUT);  pinMode(LED\_BUILTIN, OUTPUT);  } |

Initialized the variables and set the pinmode according to the purpose. **”INPUT\_PULLUP”** will add a pull up resistor in the input so if the input is not connected (button is not pressed) there will not be garbage values.

Serial.begin(115200); will set the baudrate to 115200 which is the speed of serial communication.

* **Loop:**

|  |
| --- |
| void loop() {  // put your main code here, to run repeatedly:  if (digitalRead(WIO\_KEY\_A) == LOW) {  buzzer();  ButtonA\_press\_count++;  blinking(seriesA(ButtonA\_press\_count));  }    else if (digitalRead(WIO\_KEY\_B) == LOW) {  buzzer();  ButtonB\_press\_count++;  blinking(seriesB(ButtonB\_press\_count));  }    else if (digitalRead(WIO\_KEY\_C) == LOW) {  buzzer();  //resetting both counts  ButtonA\_press\_count=0;  ButtonB\_press\_count=0;  }  //to give time so one press doesn’t registered as multiple presses.  delay(200);  } |

In loop we will have conditions on every button and a procedure it have to follow, the 200 delay is to avoid multiple presses registered.

Buzzer(); Function create a beep, then for button A and button B we increment to the counters specified for them, and pass it to create the output with respective series, then we call blink function that will make the built-in led blink x times where x in input.

* ***Functions:***

|  |
| --- |
| void buzzer()  {  //beep for 1/10th of a second  analogWrite(WIO\_BUZZER, 128); //Pwm with 50% duty cycle  delay(100);  analogWrite(WIO\_BUZZER, 0); //Pwm with 0% duty cycle  delay(100);  }  void blinking(int times)  {  //blink in a second  for (int i=0;i<times;i++)  {  digitalWrite(LED\_BUILTIN, HIGH); //LED On  delay(500);  digitalWrite(LED\_BUILTIN, LOW); //LED Off  delay(500);  }  }  int seriesA(int val)  {  //return number of blinks  int factorial=1  for (int i=2; i<=val;i++){  factorial=factorial\*i;  }  return factorial+1; //n!+1  }  int seriesB(int val)  {  //return number of blinks  return (val\*(val+1))/2; //n(n+1)/21  } |

**Results:**

The results are as expected, I got the blinking and buzzer sound according to the given series.

**Summary:**

*I* learn to work with multiple provided components of wio board, ie. Buzzer, buttons and led. I gain knowledge about Arduino language programming. And gain skills to setup WIO terminal.