**Template for Arduino and Multifunction Shield Activity**

**Statement Given:**

**Task I:**

**With codes given to you test following utilities available on multifunction shield**

1. Using Beeper on a Multifunction Shield and control the timing and frequency of beeper using MFS.beep() function which allows us to define

* On time of beep
* Off time of beep
* No of cycles of on-off
* No of loops
* Time delay between loops

1. Control Intensity of LED on Shield using analogWrite Function
2. Writing to Seven Segment Display Digits using MFS.write ()
3. Detecting Push Button Pressed using MFS.getButton()

**Task II:**

**Develop an application for**

* Setting time for Microwave ( Use 2 Pushbuttons to set Minutes (0-9) and Seconds in multiples of 10( 00-50))
* Display set time on seven segment display
* Counting down from predefined time once start is pressed ( Use Third Pushbutton as Start)
* Beep after time is Lapsed.

**Evaluation Criteria:**

1. Connections as per task given,

2. Code for Arduino for the Task.

3. Successful execution of the activities.

**Performance-15 Marks : Basic tasks = 5 marks , Activity = 10 marks**

**Submission-10 Marks**

**Team**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No** | **Roll No** | **Name** | **Work Done** |
| 1 | 16010123011 | Aaryan Dubey | Coding + Connections |
| 2 | 16010123012 | Aaryan Sharma | Coding + Connections |
| 3 | 16010123013 | Aayush Hardas | Coding + Connections |
| 4 | 16010123014 | Aayush Sawant | Coding + Connections |
| 5 | 16010123015 | Abdullah Quereshi | Coding + Connections |

**Fill your details as per following points**

* **Code for the task given**
* **Video of Actual implementation**
* **Observations , reflection on activity**

**Code for the task given:**

**Task I:**

**a)**

#include <Wire.h>

#include <TimerOne.h>

#include <MultiFuncShield.h>

void setup()

{

  Timer1.initialize();

  MFS.initialize(&Timer1);

  MFS.beep();

  delay(1000);

  MFS.beep(5,

           5,

           4,

           5,

           50);

}

void loop(){}

**b)**

int testLED = 11; // LED D4 in MF Shield

void setup()

{

  // Nothing to do here

}

void loop()

{

  for (int fadeValue = 0 ; fadeValue <= 250; fadeValue += 10)

  {

    analogWrite(testLED, fadeValue);

    delay(50);

  }

  for (int fadeValue = 250 ; fadeValue >= 0; fadeValue -= 10)

  {

    analogWrite(testLED, fadeValue);

    delay(50);

  }

}

**c)**

#include <TimerOne.h>

#include <Wire.h>

#include <MultiFuncShield.h>

void setup() {

 // put your setup code here, to run once:

 Timer1.initialize();

 MFS.initialize(&Timer1); // initialize multi-function shield library

 MFS.write("Hi");

 delay(2000);

 MFS.write(-273);

 delay(2000);

 MFS.write(3.141, 2); // display to 2 decimal places.

 delay(2000);

}

int counter=0;

byte ended = false;

void loop() {

 // put your main code here, to run repeatedly:

 if (counter < 20)

 {

 MFS.write((int)counter);

 counter++;

 }

 else

 {

 MFS.write("End");

  }

 delay(100);

}

**d)**

#include <TimerOne.h>

#include <Wire.h>

#include <MultiFuncShield.h>

void setup() {

 // put your setup code here, to run once:

 Serial.begin(9600);

 Timer1.initialize();

 MFS.initialize(&Timer1); // initialize multi-function shield library

}

void loop() {

 // put your main code here, to run repeatedly:

 byte btn = MFS.getButton(); // Normally it is sufficient to compare the return

 // value to predefined macros, e.g. BUTTON\_1\_PRESSED,

 // BUTTON\_1\_LONG\_PRESSED etc.

 if (btn)

 {

 byte buttonNumber = btn & B00111111;

 byte buttonAction = btn & B11000000;

 Serial.print("BUTTON\_");

 Serial.write(buttonNumber + '0');

 Serial.print("\_");

 if (buttonAction == BUTTON\_PRESSED\_IND)

 {

 Serial.println("PRESSED");

 }

 else if (buttonAction == BUTTON\_SHORT\_RELEASE\_IND)

 {

 Serial.println("SHORT\_RELEASE");

 }

 else if (buttonAction == BUTTON\_LONG\_PRESSED\_IND)

 {

 Serial.println("LONG\_PRESSED");

 }

 else if (buttonAction == BUTTON\_LONG\_RELEASE\_IND)

 {

 Serial.println("LONG\_RELEASE");

 }

 }

}

**Task II:**

#include <TimerOne.h>

#include <Wire.h>

#include <MultiFuncShield.h>

int set\_counter\_min = 0;

int set\_counter\_sec = 0;

int remaining\_seconds = 0;

void setup() {

Serial.begin(9600);

Timer1.initialize();

MFS.initialize(&Timer1);

MFS.write("CONT");

delay(2000);

MFS.write("ON");

delay(2000);

MFS.write("SET");

delay(2000);

}

void loop() {

byte btn = MFS.getButton();

if (btn) {

byte buttonNumber = btn & B00111111;

byte buttonAction = btn & B11000000;

Serial.print("BUTTON\_");

Serial.write(buttonNumber + '0');

Serial.print("\_");

if (buttonAction == BUTTON\_PRESSED\_IND) {

Serial.println("PRESSED");

if (buttonNumber == 1) { // Button 1 increments seconds

set\_counter\_sec++;

} else if (buttonNumber == 2) { // Button 2 increments minutes

set\_counter\_min++;

} else if (buttonNumber == 3) { // Button 3 starts countdown

MFS.write("SET");

MFS.write("TIME");

// Calculate total seconds

remaining\_seconds = set\_counter\_min \* 60 + set\_counter\_sec;

// Display the set time

MFS.write((int)set\_counter\_min);

delay(1000);

MFS.write((int)set\_counter\_sec);

delay(1000);

// Countdown loop

for (int i = remaining\_seconds; i > 0; i--) {

set\_counter\_min = i / 60;

set\_counter\_sec = i % 60;

MFS.write((int)set\_counter\_min);

delay(1000);

MFS.write((int)set\_counter\_sec);

delay(1000);

}

// Beep after the countdown

MFS.beep(500); // Adjust the frequency and duration as needed

MFS.write("End");

set\_counter\_min = 0;

set\_counter\_sec = 0;

}

delay(100);

}

}

}

**Video of Actual Implementation:**

<https://drive.google.com/drive/folders/149pE3_O0yyovkHLBFF0orxtLToeNGin0>

* **Observations, reflection on activity :**

We learnt about Multifunction Shield-how to implement it on the Arduino kit and getting the desired output.We Have successfully performed the activity