User Guide

1 Using the application

1.1 Overview

The Device for Examining Multisensory Exploration along with the accompanying Windows Application is a tool that enables the user perform experiments where human subjects are exposed to a series of sensory stimuli and their reactions are recorded.

The device is able to continuously monitor the acceleration, angular velocity and angle in the three directions and send the readings to the application. The device is able to detect a set of events such as: button being pressed, loud sounds such as knocks on the device of user's reactions, device being shaken along the x,y and z axes, device rotation about he x, y and z axes and device being squeezed along the x, y and z directions. These events are detected by the corresponding sensors and a notification is transmitted to the Windows application.

The device can also apply stimuli to the user. This can be done at any time from the PC application or as a response to a detected event. The device is capable of applying the following types of stimuli: play a certain sound file, play a specific vibration pattern and displaying LED patterns on each of the 5 LED matrices.

The incoming IMU readings are added in the Inertial Measurement Unit Readings text box while detected events and sent responses are logged in the Action and Response Event Log text box. The acquired readings and event log can be exported as an XML file. The application also offers a Pattern Designer applet where custom LED patterns can be created.

1.2 How to turn the device on and off

In the normal mode of operation the device is powered on by pressing the button located on the upper right side of the USB port. Note that a pin is needed to access the button which was buried below the surface of the panel in order to restrict the user's interaction with it during the experiments. A red LED on the left of the button will indicate that the device is powered on. Pressing the power button again will power off the device and the LED indicator.

1.3 How to charge the device

In order to charge the device, connect it to a 5V power supply via the Micro USB port. Please ensure that the power switch is in the ON position otherwise the device will not charge as the battery is not connected to the Arduino. When charging, an amber glow from the charging pin on the Arduino can be seen to the right of the USB port.

1.4 How connect the app to the device and start the experiment

After the device is powered on, a WiFi access point will become visible to devices with WiFi capability. The name of the network is ESP_86E2F5 and it does not require a password. After connecting to the WiFi access point, the device can be controlled using the Windows Forms Application.

After opening the Windows application the device should appear as "Not connected". After pressing the connect button the status should change to "Connected". If an error message appears and the status does not change please check that the computer is connected to the WiFi access point.

Pressing the START button next to the EXPORT DATA button will start transmission of data from the device to the PC. Periodic entries with the positional data acquired from the device will be added to the text box under "Inertial Measurement Unit Readings". If the device detects an event it will be logged in the "Action and Response Event Log". Pressing the STOP button will pause data transmission from the device to the PC.

1.5 How to select automatic and manual responses

Automatic responses are selected from the "AUTOMATIC RESPONSES" area from the main window of the application. Clicking on one of the check boxes in the list will enable the "Select response type" combo box. There the type of response (play sound, play haptic effect, display LED pattern) can be chosen. Depending on the choice, the "Select sound file", "Select haptic pattern" or "Select matrix" and "Select pattern" combo boxes will be enabled. After desired response is selected, the trigger action in the checked list will be marked as checked and the selected response will be shown right to the checked item. If the response has to be changed, the described procedure should be followed again.

Manual responses are triggered from the "MANUAL TRIGGERED RESPONSES" box. There the user can select directly the desired action from the combo boxes. Pressing the START button on the left of the combo box will send the selected action to the device.

1.6 How to export the data

Pressing the EXPORT DATA button will create a XML file and open a save file dialog from which the user can select the location where the file will be saved. The file contains two tables, Table 1 with IMU readings and Table 2 which has the event log entries.

2 Advanced configuration

To update the software running on the Arduino, the user has to connect the device to a computer through the micro-USB port. Then, using the Arduino IDE, the C/C++ code which can be found at: (github) can be modified and uploaded to the board. When operating in the normal mode, the device sends status messages via the USB connection. These can be investigated by opening a Serial Monitor in the Arduino IDE (Tools -> Serial Monitor) and setting the baud rate to 115200.

2.1 Adding more sound files

In order to add more sound files first the main PCB inside the device must be accessed. This can be done by unscrewing the screws on the main face and pulling the face with the attached PCB slowly outside while carefully disconnecting the 4 pin connectors attached to the other 5 faces. The soundboard's micro-USB port can then be connected through a USB cable to a computer. Note that the switch under the soundboard should be pushed towards the edge of the PCB from the centre position. The soundboard will appear on the PC as a flash drive and sound files with the right format (OGG or WAV) can be uploaded to the board by simply copying the files to it. The files should then be renamed so that all files on the soundboard follow the naming sequence (T00, T01, T02, etc.). After the files are uploaded, the USB cable can be disconnected and the switch under the soundboard should be switched back to the middle position.

The list of sound files from the Windows application should be uploaded with the names of the newly added sound effects so that they can be triggered from the application. This list is a text file "Soundfiles.txt" that can be found in the application folder at: `MultisensoryDevice-BEngProject\bin\Debug`. In the file, T00 should correspond to the text on line 1, T01 to the text on line 2 etc.

2.2 Adding more led patterns using the Pattern Designer

More patterns can be created using interactive Pattern Designer applet. This can be deployed by pressing the "Pattern Designer" button from the top menu.

The user can select form a combo box the desired colour from the available list of 14 colours and proceed to press the buttons corresponding to the NeoPixels that will display that colour. When the button is pressed, it will change colour and a line of code which turns on the pixel in that position with that colour will be appended to the text box on the right. The user can change the colour at any time and continue drawing. If the pattern has to be changed, pressing the coloured button again will delete its corresponding line of code and will revert it to its initial state. The whole form can be reset and a new pattern drawn by pressing the "Reset" button.

After drawing the pattern, the user can write a name for the pattern in the single line text box and then press the "GENERATE Pattern" button. This will replace the preset name "NEWPATTERN" with the desired name and will generate the complete code of a function is ready to be copied to the Arduino program and loaded on the microcontroller.

The copied code should be pasted at the end of the Arduino file among the other pattern functions. After that, an extra case command should be added to the switch function form the display_pattern() function shown below. The case should follow the format of the ones already there, calling the newly created pattern function.

```
void display_pattern(Adafruit_NeoMatrix matrix, int pattern)   //function that selects the
    required pattern
{
  matrix.begin();
  matrix.fillScreen(0);          //turn off all LEDs
  matrix.show();
  switch (pattern)               //call the function corresponding to the selected pattern
  {
    case 1: patternRED(matrix); break;
    case 2: patternGREEN(matrix); break;
    case 3: patternBLUE(matrix); break;
    case 4: patternYELLOW(matrix); break;
    case 5: patternSMILEYFACE(matrix); break;
    case 6: patternRO(matrix); break;
    case 7: patternSCO(matrix); break;
    case 8: patternRECTANGLES(matrix); break;
    case 9: patternLINES(matrix); break;
    case 10: patternGREEN(matrix); break;
    //==================== INSERT CASE HERE =============
  }
}
```

An extra step should be done in order to use the created pattern from the Windows application. A new entry should be added to the text file "Matrixpatterns.txt" which can be found in the application folder at `MultisensoryDevice-BEngProject\bin\Debug`. The line in the text file should correspond to the number after the case command added above. After uploading the new Arduino program and opening the Windows application, the new pattern should be available and ready to use.

2.3 Tuning sensor sensitivity and timing

The definitions shown in the code below from the header of the Arduino file define the sensor threshold values and timing constants. These values can be changed and the new program uploaded to the Arduino.

```
#define SHAKE_THRESH 1.2    //shake detection threshold <g>
#define ROT_THRESH 170      //rotation detection threshold <deg/s>
#define FORCE_THRESH 25     //force detection threshold
#define IMU_PERIOD 500      //IMU readings sending period <ms>
```