

SHIMMER 3

Alert button



Written by Steffan Lildholdt (steffan@lildholdt.dk)

Version 1.0

TABLE OF CONTENT

1. Introduction.....	1
2. Firmware architecture.....	2
3. Firmware design	3
4. Testing the firmware	4
5. Modifying the firmware	5

1. INTRODUCTION

Shimmer is a small sensor platform well suited for wearable applications. The integrated kinematic sensors, large storage and low-power standards based communication capabilities enable emerging applications in motion capture, long-term data acquisition and real-time monitoring. Shimmer offers the possibility for custom firmware integration so that the device can be tailored to fit many different applications. Shimmer3_AlertButton is an example of customized firmware.

Most seniors want to remain living in their own home as long as possible. If they live by themselves and an accident occurs in the home it is comfortable knowing that help is close and all it takes is a push on a button. Currently there are many devices that are able to offer this kind of functionality. These are sometimes called Personal Emergency Response System, Medical Alert, or Medical Emergency Response Systems. This piece of firmware, for the Shimmer 3, resembles those systems although being simpler.

When the user pushes the button on the Shimmer 3 a Bluetooth message is sent to a nearby computer. Along with this firmware a simple C# console application has been developed which listens for the button command from the Shimmer 3.

Section 2 and 3 describe the architecture and design of the firmware so an overview can be obtained quickly. Section 4 is about how to test the firmware along with expected outputs.

2. FIRMWARE ARCHITECTURE

The software architecture of the alert button firmware is a combination of layered and module based and is illustrated in Figure 2.1. Each element is explained in the following.

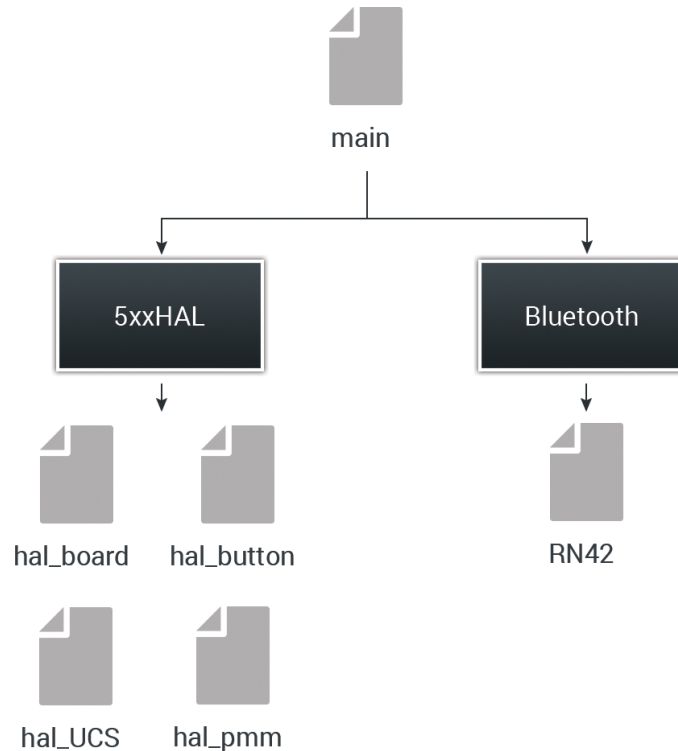


Figure 2.1 – Software architecture of fall detection firmware

Main

The main file is responsible for initializing all external components and controlling timers. Due to limited data processing all of this has also been placed in the main file.

Bluetooth

This is a driver for the RN42 Bluetooth module.

5xxHAL

A collection of various helper files. Hal_board is used to setup the board and control LEDs. Hal_button is a driver for the button. Hal_pmm control the power consumption (Power Management Module). Hal_UCS is driver for the timers (Unified Clock System).

3. FIRMWARE DESIGN

An overview of the alert button algorithm is illustrated in the activity diagram in Figure 3.1.

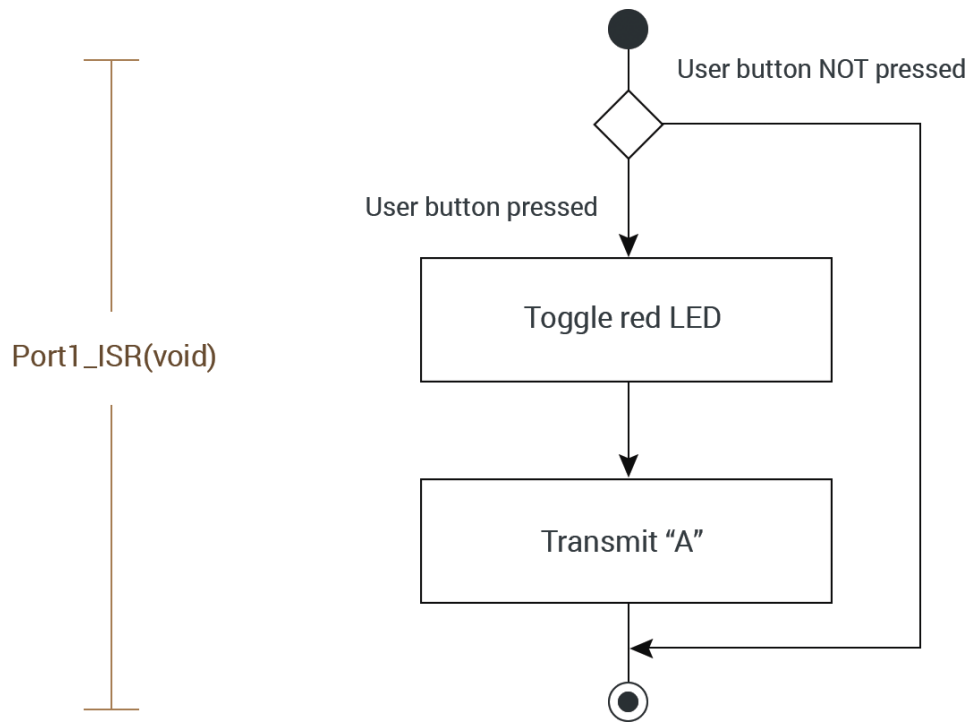


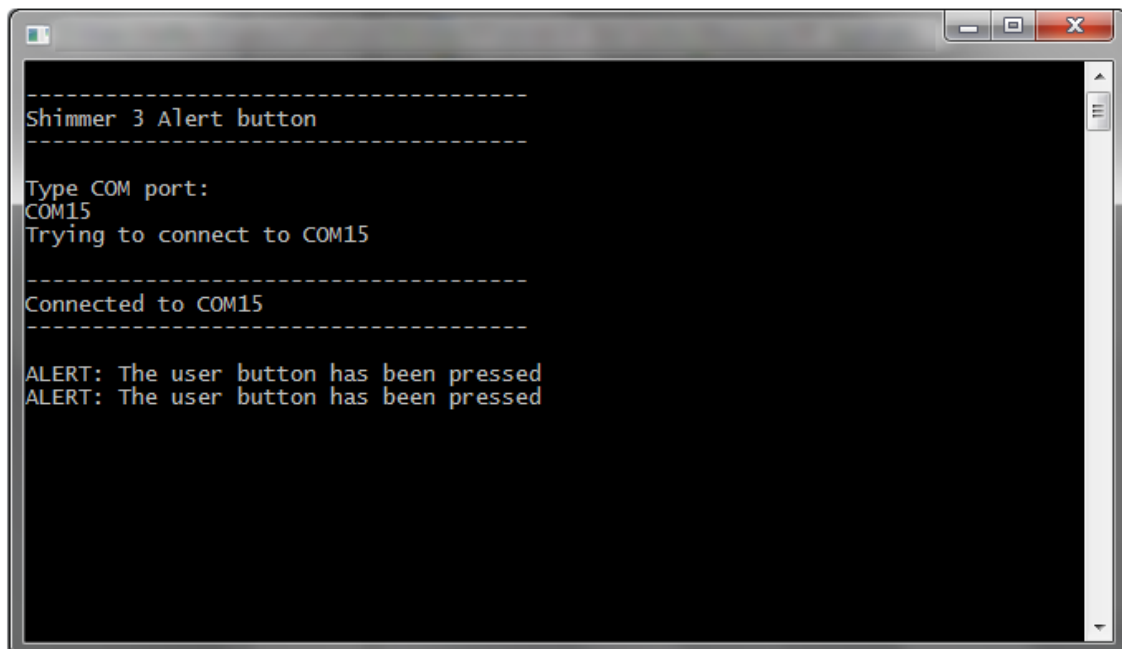
Figure 3.1 – Activity diagram of the Alert button firmware

When the user button on the Shimmer 3 is pressed the Interrupt Service Routine of Port1 is called. This function first toggles the red LED and then transmit the character "A" via Bluetooth.

4. TESTING THE FIRMWARE

To test the firmware, follow these steps.

1. If the Shimmer3 is not added as unit on your computer then follow the instructions in this link <http://windows.microsoft.com/en-us/windows7/add-a-bluetooth-enabled-device-to-your-computer>
2. Load the bootstrap "Shimmer3_AlertButton.txt" onto the Shimmer. The file is located in the folder "Resources/Firmware".
(How to load custom firmware onto the Shimmer is described in tutorial 1)
3. Launch the PC application "Shimmer3_PCApp" located in the folder "Resources/PC Application"
4. Enter the COM port of the Shimmer Bluetooth connection (i.e. COM15).
5. The program should display the text "Connected to COMxx" and the blue LED on the Shimmer should be on.
6. When the button is pressed on the Shimmer 3 the program should display the text "Alert: The user button has been pressed".

A screenshot of a Windows application window titled "Shimmer 3 Alert button". The window has a black background with white text. The text inside the window is as follows:

```
-----  
Shimmer 3 Alert button  
-----  
Type COM port:  
COM15  
Trying to connect to COM15  
  
-----  
Connected to COM15  
-----  
  
ALERT: The user button has been pressed  
ALERT: The user button has been pressed
```

The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Figure 4.1 - The program output at successful connection and button press

5. MODIFYING THE FIRMWARE

What happens when the button on the Shimmer is pressed is located on 115-127 in the firmware code:

```
//BUTTON_SW1
case P1IV_P1IFG6:

// Toggle red LED to indicate button press
Board_ledToggle(LED_RED);

// If Bluetooth connection is enabled transmit the buffer
if(btIsConnected)
{
    BT_write("A", 1);
}

break;
```

When the button is pressed the red LED toggles and a single character ("A") is transmitted. To modify this just replace the content of **case** P1IV_P1IFG6.