REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Implementation

1.1 Our device description

We had two boards. One board recorded pitch and yaw and sent it wirelessly to the other board. The second board did most of the work. It had an LCD and a keypad connected to it. It would play one of two audio synthesizers (square wave or frequency modulation). The keypad would control the volume and pitch of the sound. It would also allow toggling between synthesizers. The pitch and yaw recieved from the wireless chip was used to control certain effects applied to the sound synthesised: the pitch would control resonance and the yaw would control low frequency cutoff. The LCD display would display the volume, the note, the synth mode and the pitch/yaw.

1.2 Development process

The required structure of this project shared much in common with lab 6 so we decided to use that as a starting point. After modifying the main program loop from lab 6 to suit our needs, we worked on configuring the peripherals and reading from/writing to them. We then needed to get a double buffered DMA stream writing to the DAC. Our first designed had timing issues as the time it took to set the LCD screen was longer than the time to write an audio buffer. Our first desgined had all periferal IO and audio synthesis occur on a single loop iteration while the DMA wrote a buffer to the DAC. Since the time to write the LCD was longer than the time taken for the DMA to write a buffer, the program couldn't keep up and the DMA would crash. To remedy this, we designed a second program that put the LCD writing in it's own thread. The value to be written to the LCD were updated on every loop iteration but the actual printing occured at it's own pace.

1.3 Program Flow

1.3.1 First Board

The first board just pumped out pitch and yaw data to the wireless chip. It was quite simple and consisted of two threads: one for the main loop and accelerometer calculation and the other for wireless transmission. The program flow is as in Figure 1.

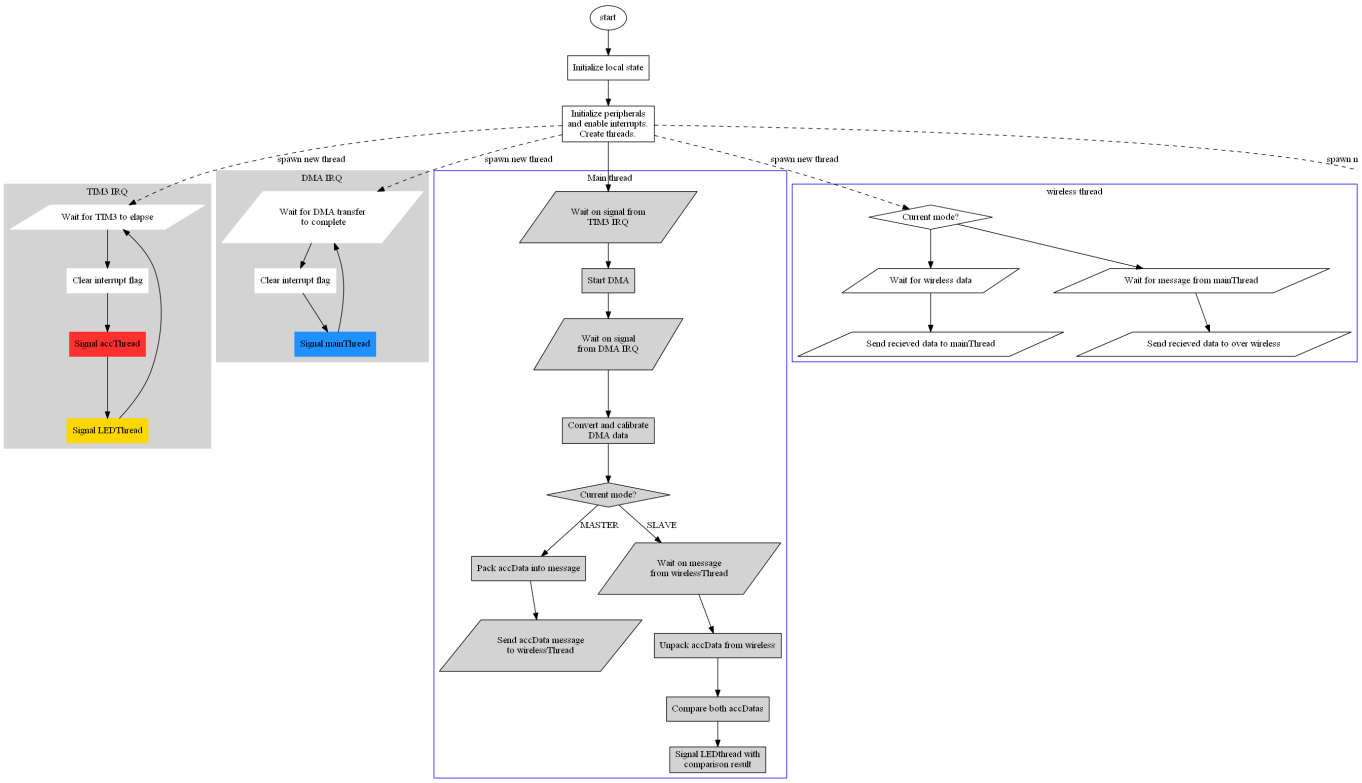

Figure 1: First Board Program Flow

1.3.2 Second Board

The second board is in charge of most of the work. It has three regular threads and 3 interupt handlers.
The threads are:

- main (generate audio and read keypad)
- wireless
- LCD

The program flow, with the interrupt handlers removed, is as in Figure 2.

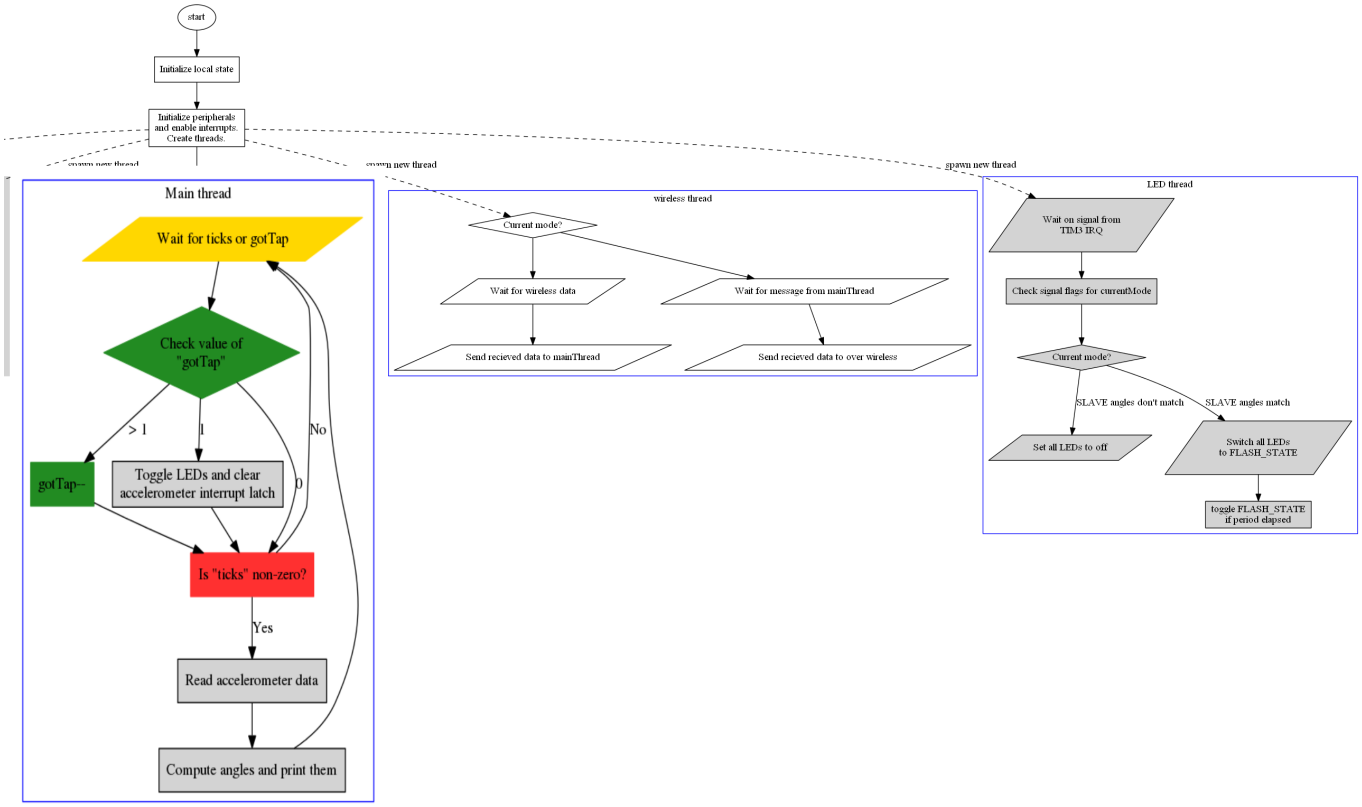

Figure 2: Second Board Program Flow

We used keil's event recording mode to ensure that no threads were busy waiting. As can be seen in Figure 3, most of the time is spent in idle mode so the threads are yielding properly.

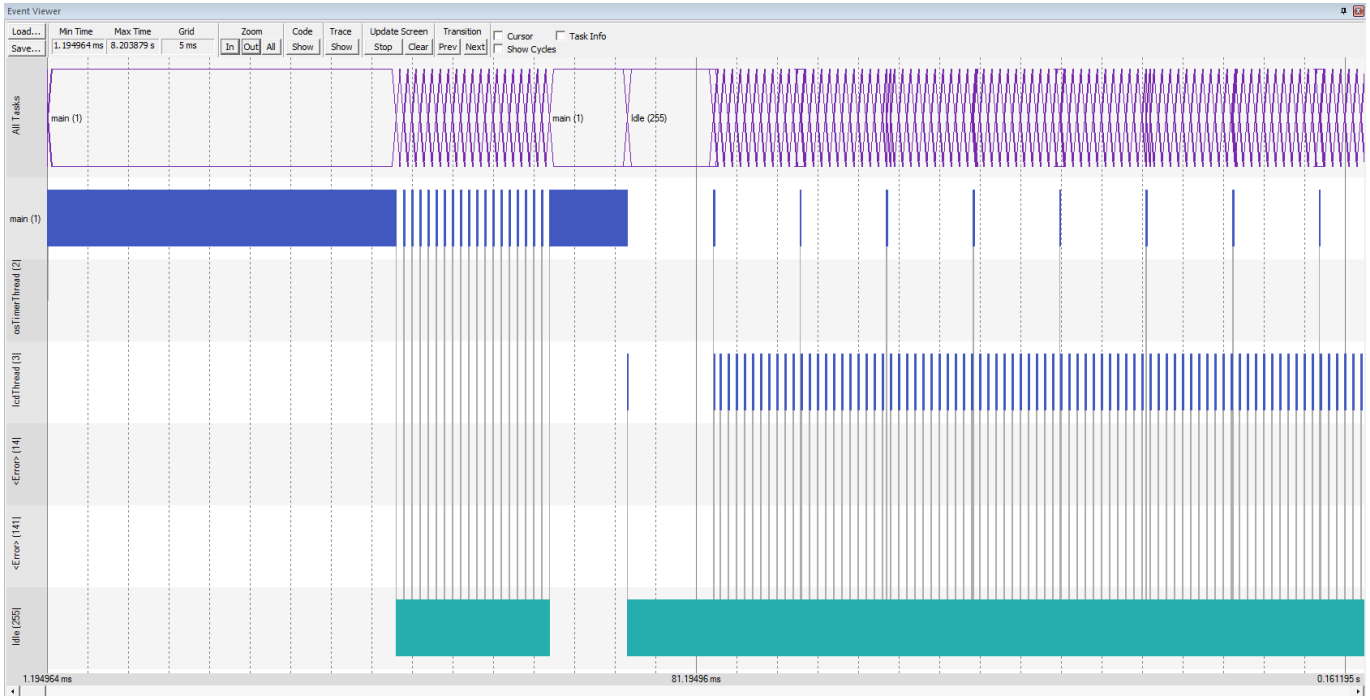

Figure 3: Event Viewer recording of our program running