**Nikolay Stepin**

**Dominick Przybylo**

**University of Illinois Chicago**

**ECE 266 – Introduction to Embedded Systems**

**Fall 2017**

**Group 11, Tuesday 10:00-11:50**

**Project Progress Report II**

**MotionIT\_MP3**

**I. Introduction:**

We are building a relatively simple motion-sensing wireless MP3 player. This system can be reorganized into a distributed wireless alarm system. The idea is simple; however, it serves as a creative way to expand on our knowledge of embedded systems and get exposure to coding and working with wireless technologies/standards, as well as serial communication protocols.

**II. Requirements Form:**

Name: MotionIT\_MP3

Purpose: Consumer-grade wireless MP3 player for leisure/distributed home alarm system

Inputs: Movement, push-buttons on the launchpads

Output: Speaker/headphones interfaced to the MP3 module

Functions: Wireless data transmission, MP3 playback from SD cards

Performance: plays music/alerts the tenant within 5 seconds of motion detection

**Cost:**

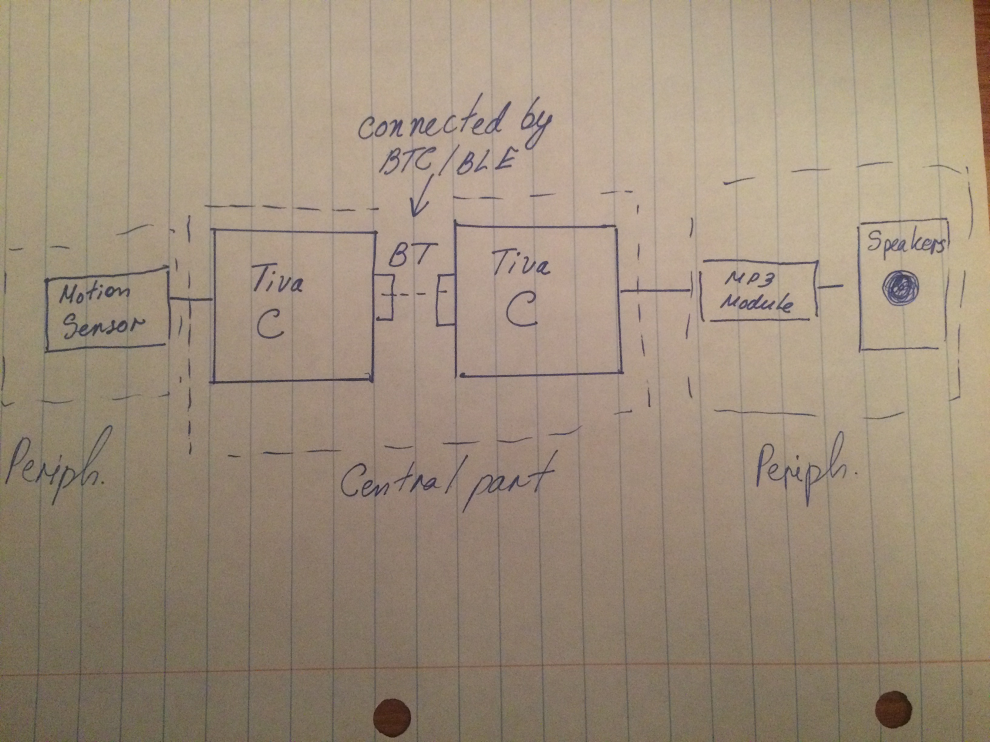
Component Cost: $75 without the launchpads, ~$105 with launchpads (if purchased)

The total cost ended up being about $130, however, because we also ended up purchasing a new SD card, as well as 2 extra Bluetooth modules.

If the unit was to be considered for production, then it would just be the Tiva C MCUs with a controller/motion detector, as well as the MP3 part. The MP3 part would probably contain the same MCU as the MP3 module does, but be placed on the same PCBA as the Tiva C MCU is on. This would be, most probably, more expensive than the typical standalone MP3 player, but as a wireless MP3 player that can connect to different slave nodes, as well as smartphones, it could be somewhere in the average range, given that the PCBAs are done correctly as to make essential functions possible while removing all the unneeded functions.

**III. Architecture:**

The system architecture can be pictured in the following way:



The hardware architecture:qq

Motion Sensor

TX/HM-10 MCU

Tiva CPU 1

qQqqqq

qqqqq

qqqqeqqeqeqee

RX/HM-10 MCU

Tiva CPU 2

qqqq

MP3 Module/KT403A MCU

SD Memory

Speakers

Software architecture:

GPIO Driver to MCU

Motion

TX/RX UART Connection

Hex UART Messages for MP3 Player

Sound

**IV. Project Summary:**

We developed the MotionIT\_MP3 and have made a functional prototype of the system that involved 2 Tiva C launchpads, 2 Bluetooth modules, and 1 MP3 module. The Bluetooth drivers were written with the help of the UART library for the Tiva C launchpad. The principle is that the AT commands that are inherent to the HM10 MCUs can be sent as strings through the UART to the right port and UART RX/TX pins to configure the modules. That is exactly what we did, by creating two initialization functions—one for RX and TX sides respectively. The MP3 module was very similar, except it receives hex commands to configure and use the module through its on-board KT403A MCU. On the system level, we’ve managed to implement motion-detected transmit char which, when received, starts playing songs in a specified folder at the index of 1 (that is configured through the commands sent to the module). We’ve implemented play and pause features, as well as play previous song and play next song feature.

The project was hard not because the concept was hard, but because of the lack of information. For the Bluetooth modules, the data sheet provided by the company that sold the two initial modules that we have was half-translated (i.e. the parts that the company thought are not needed were left out), as well as some set-up parameters were wrong (e.g. the baud rage). Therefore, we had to google the chip that is on the module and go from there, looking up UART and Bluetooth solutions that use the same chip. The same occurred with the MP3 player, where I had spent considerable time wondering why it is not working, although my hex commands seem to be right and why I’m getting a weird message over UART when I monitor the UART register. After googling that returned command, I made substantial progress very fast, but It was not a good thing that error codes were not included in the original data sheet and we had to use a different data sheet from another company which also makes (copies) modules with the same MCU chip. We also learned how to utilize RealTerm hyperterminal program, and how to test embedded modules on the PC, before implementing drivers on the Tiva C MCU.

In addition, this is a rough prototype. I believe there are areas on which the project could be improved on, especially timing and power consumption. One of the most inefficient design choices in our project was the fact that we used polling and not interrupts for both the GPIO and the UART transmission. We have studied GPIO interrupts in the lab, so that wouldn’t be hard; neither should be the UART interrupts, as even though we did not cover them in the lab, they function the same way. It would also help with getting out of an infinite loop with the uartCharGet() function, which waits until it receives a char preventing the MCU to work and process other requests/functions. We also didn’t implement some other MP3 functions, such as changing the volume. We thought of adding the rotary actuator, as that would make sense (similarly to how the volume is controlled in typical speaker-set) and it would be an easy way to change the volume.

**V. Conclusion:**

During this project, we’ve had a taste of how it would be like to develop a project in the real-world, because both the Bluetooth modules and the MP3 module were new modules that we’ve never worked with. Therefore, despite understanding the concept behind their operation, we had to figure out the details as we went with the project. It was a great learning experience and by focusing on the wireless aspect, we were able to step into the world of IoT, even if the step is small.

The responsibility for this project was divided as follows:

Nikolay:

- Came up with the initial idea for the project.

- Wrote majority codes for the necessary drivers for the project

- Revised codes after debugging to fix minor errors.

- Wrote majority of the report

Dominick:

- Worked on the mp3 driver files for the project

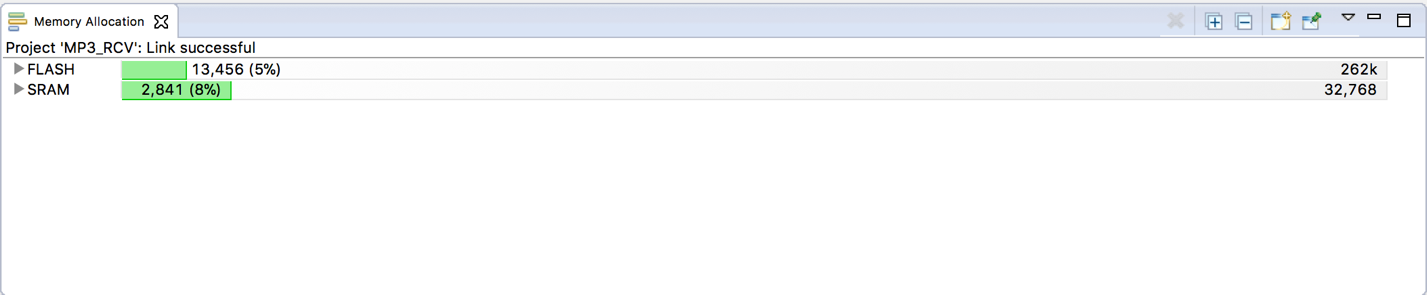
- Revised some of the codes

- Created the project folders

- Revised the report and added a few things

SRAM and flash memory for each side:

MP3\_RCV



MP3\_TX

