Middle East Technical University

EE447 Term Project

Preliminary Report

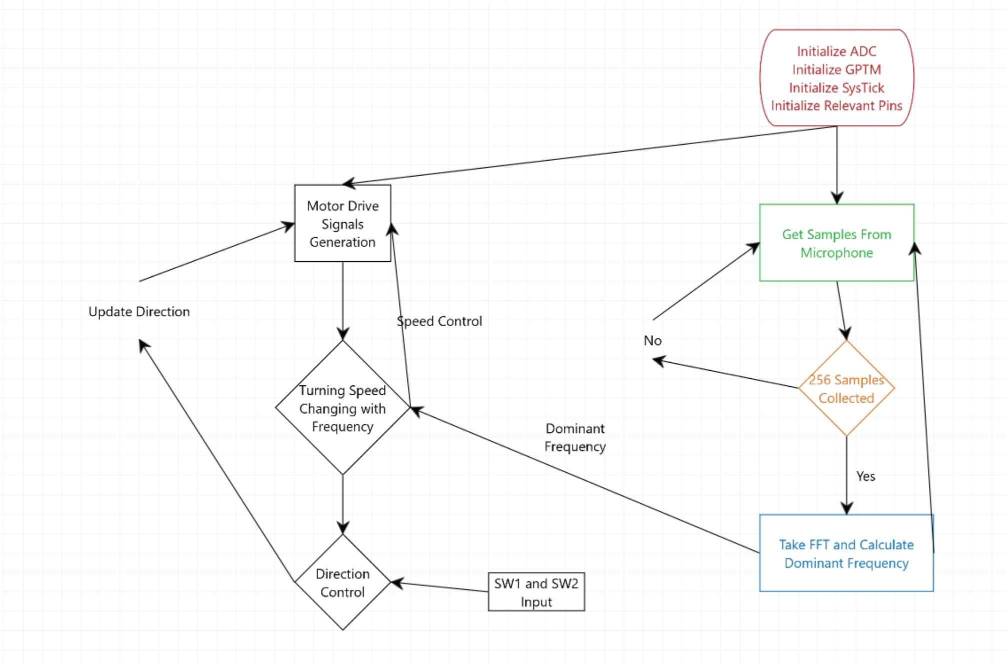
Emrullah Çelik

2307320

System Description

This project consists of different subparts which rely on each other. Roughly one can divide the project into three submodules:

1. Sampling the sound and finding the dominant frequency
2. Driving the motor and changing led color according to detected frequency.
3. Displaying threshold and frequency values on Nokia 5110 screen.



For motor and audio sampling, the flowchart is given. The display module will display values and update them every 1 second.

Pseudo Code for main

Initialize all ports and submodules

Pool SW1 and SW2

Write rotation direction to memory

If more than 1 sec passed update the display

Submodule 1 – Sampling the Sound

In this module by using GY-MAX9814 microphone sound samples will be collected at a frequency of 2KHz. These samples will be stored in memory as an array. When the array length becomes 256, then the Fast Fourier Transform (FFT) function from the DSP library will be called. DSP library will take the Discrete Fourier Transform of the sound samples and write it over the sound samples. So, the operation is performed in place.

To achieve sampling at 2KHz frequency interrupt will be used. Specifically speaking, the System Tick counter will be used to generate an interrupt. When the System Tick timer counts to 0 it will invoke an interrupt. The Startup.s file will be modified so that an interrupt subroutine for sound sampling will run when that interrupt is invoked. This subroutine will use the ADC module to sample analog signals and convert them to digital.

When 256 samples are collected and written to memory DSP libraries FFT function will be called. Initially, values from “arm_cfft_sR_q15_len256” will be given, the start address of the sample array will be specified and some registers’ values will be assigned. After that “arm_cfft_q15” subroutine will be called. This subroutine will take the FFT in place. Lastly, the dominant frequency will be found from the FFT result. To achieve that objective magnitude of the complex numbers will be calculated and compared. Complex the number with the highest magnitude is our dominant frequency and its index will be used at finding the frequency. To find the frequency formula 1 will be used.

$$\frac{Sampling\ Frequency}{\#\ of\ Samples} * Index = Frequency\ at\ given\ index$$

Since the index will be the index of the highest magnitude number from the result of the FFT, it will give us the dominant frequency. This frequency information will be used in other submodules.

Pseudo Code

Initialize ADC

Initialize Pin

Initialize SysTick

Initialize counter to 0 and write it to memory

In interrupt subroutine

Get a sample from ADC pin

Shift data 4 byte left

Save data to memory

Update counter and write it to memory

If(counter == 256)

Call FFT subroutine

Counter <- 0

In FFT subroutine

Take FFT

Calculate dominant frequency

According to dominant frequency, update LED color

According to dominant frequency update motor speed

Submodule 2 – Driving the Motor

To drive the motor periodic drive signal needs to be generated. For that reason, another interrupt subroutine will be used so that drive signals can be generated at a constant rate periodically. In order to generate interrupts for driving the Motor GPTM will be used. As in the previous submodule, when the timer counts down to 0, it will generate an interrupt. The contents of Startup.s will be modified so that when that interrupt occurs it will call the subroutine which will be generating the required signals for driving the motor. The speed of the motor will change according to the frequency detected. When the frequency is in the low range motor will turn slowly. When it's in the middle range it will turn at a middle speed and when it is in the high range motor will turn fast. Also, the color of the RGB LED that is placed on the TIVA board will change its color according to frequency. For low, mid, and high frequency; red, green, and blue light will be emitted, respectively.

To change the direction of the motor SW1 and SW2 that are mounted on the TIVA board will be used. For SW1 PF4 and for SW2 PF0 pins are going to be used. PF0 is locked, therefore it will be unlocked before usage. When one of the buttons is pressed it will change the direction of rotation when it is released. To pass that information to the motor driving interrupt, I am planning to use memory. I will store rotation direction to a memory word. When a switch is pressed, it will modify that memory space. The motor driving subroutine will look at that memory for the direction of rotation.

Pseudo Code

Initialize GPTM

Initialize GPIO

Write initial rotation direction to memory

In subroutine

> Read memory address where rotation direction is stored.
>
> Create a 4-bit driving signal for the given direction
>
> Clear ICR register for new interrupts.

Submodule 3 – Displaying Information

On the display, upper and lower frequency, and amplitude threshold values will be displayed. Also, current frequency and amplitude values will be visible on the display. The communication will be done via SPI protocol.

At the first step, the display should be initialized correctly. After initialization, commands and write orders can be sent. In order to distinguish commands and write orders, a D/C pin is used. If D/C is 0 it's a command, if it's 1 then it is a write order. In order to display text on screen bit maps will be used.

For example, in order to write "E" to the screen, we will generate a bitmap of it and pass each column.

So 0xFF 0xFF 0xDC 0xDC 0xDC will print "E" to the screen.

The location of the cursor will be set by command options.