CS 684: Embedded Systems

Project: FIR BOT

Team 12 Members
Arjun Joshi (10307925)
Prashant Bhut (10307036)
Manish Patidar (10307015)

1. Introduction

Today society is witnessing a rapid rise in crimes in major cities. In countries like India, there are not enough police personnel to safeguard each and every citizen due to vast population. Hence we need a system which helps the law enforcement agencies to quickly identify perpetrators of crime and bring them to justice. With this thought in mind we have attempted to prototype a system which detects sound of gunshot and immediately triangulates the position of the shooter and captures the person on camera. This system will upload the image to a central station and the police can take swift action from thereon.

2. Problem Statement

In our prototype system we have employed sensors in square grid pattern around our system. Any gunshot from any direction will be picked up by the sensors and the direction of sound will be triangulated. The firebird robot will move in the triangulated position and capture the image of the person who fired the shot and upload it to PC.

3. Requirements

3a. Hardware:

1. dsPIC33FJ128GP802 microcontroller
2. Firebird Robot
3. LM324 amplifier
4. PICkit3 programmer
5. Microphone Sensors
6. MAX232 for RS232 communication
7. Zigbee module
8. USB Camera

3b. Software

1. Matlab code for filter simulations
2. BandPass implementation of filter in dsPIC
3. MPLab IDE for dsPIC programming
4. AVR studio for Firebird programming
5. Matlab code for zigbee communication between FireBird and PC
6. Matlab code for capturing image through camera
7. Triangulation logic implementation in FireBird

4. Implementation

4a. Hardware

1. Circuit design for interfacing dsPIC and sensors using LM324 amplifiers
2. Circuit for interfacing dsPIC and Firebird via MAX232.

4b. Software

1. Filter simulated in Matlab. Frequency of interest found using Matlab and toy gun
2. Matlab code for zigbee and camera interfacing implemented
3. Code for configuring UART, ADC in dsPIC implemented
4. Code for filter implemented in dsPIC
5. Triangulation logic implemented in FireBird
6. UART, Zigbee communication implemented in FireBird

5. Testing Strategy

5a. Filter

The band pass filter was first implemented in Matlab to identify the frequencies of interest and obtain filter coefficients. The gun was used in closed room with no source of noise and passed through the filter and the output was observed. Then the gun was again fired in presence of noise and the data was passed through the filter. The noise was filtered out.

5b. Sensor Calibration

The sensors had to be calibrated first individually to find a common threshold above which we can consider a gunshot. We first bunched the sensors together and fired a single shot. It was observed that the sensitivity of each sensor was vastly different and we found huge difference between each sensor value for same gunshot. So we had to go for less optimised way by firing multiple shots for each sensor and averaging them out.

5c. Communication Path

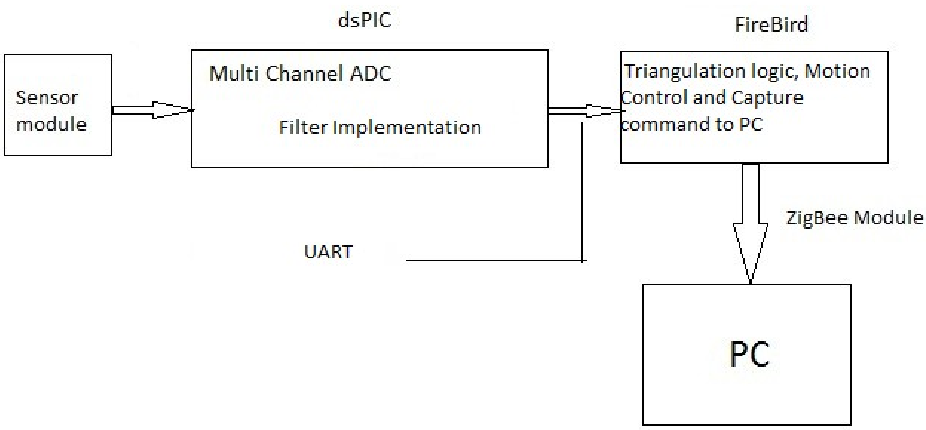
The dsPIC is a very complicated controller with multiple peripherals. It was a challenge to configure ADC for sampling 4 channels simultaneously. To check the ADC configuration for 4 channel we sent all the ADC data through UART on the PC. Since we had to implement UART for communication with Firebird it enabled us to check the communication path.

5d. Firebird Movement

We had to calibrate the movement of the firebird after triangulating the direction. We calculated that it took approximately 565 ms for the firebird to move for an angle of 90 degree. This information was necessary to stop the firebird after moving a certain angle.

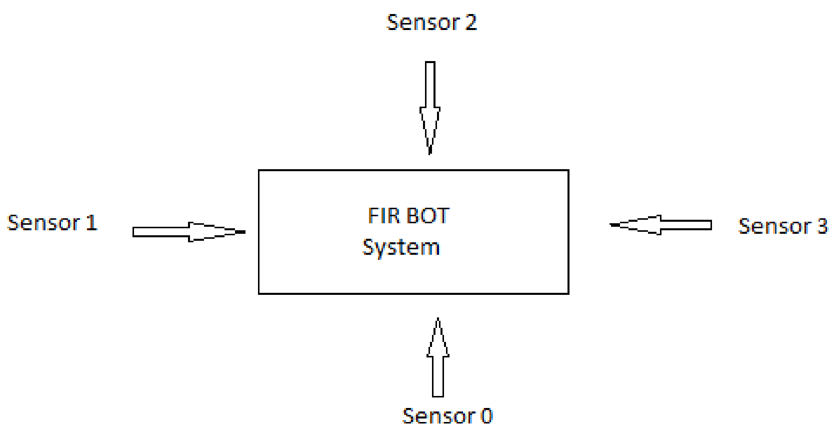
6. System Discussion

The overall block diagram of the system is shown below:



The sensor module is interfaced via an amplifier circuit to the dsPIC controller. The dsPIC controller is configured to sample all the sensor data simultaneously via ADC. Since the ADC in dsPIC only has one buffer we used the DMA engine available to store the sampled data. When ADC converts the analog data into digital it interrupts the DMA and DMA stores the ADC ouput in a DMA RAM. All the configuration is done in the source file for dsPIC (dspic.c). 32 samples for each data are collected and convoluted with filter coefficients obtained from the Matlab simulations. The filtered output is squared and averaged by the number of samples to obtain a single average value of all the samples.

If any of the average value for each channel is above a common threshold then it is confirmed that a gunshot is fired. All these channel values are sent to Firebird over UART. Once the Firebird receives all the 4 channel data it stops the UART from receiving any other data. The arrangement of the circuit is shown below.



Once the data is received, we find which sensor data is maximum among the 4. Then we find data of 2 sensors nearest to it. For example if sensor 0 had max data we compare data of sensor 3 and sensor 1. We then apply logic to see where the firebird has to move either to the right or the left. We calculate the angle to move using the 2 sensor values obtained earlier. Using the calibrated delay of firebird to move 90 degrees we move the firebird in the required direction. At this point we send a command ‘C’ to PC via Zigbee module to caputure the image of the person.

There is Matlab code running on the PC. When ‘C is received to the Zigbee module connected to the PC, PC sends a command to the USB Camera to capture the image and we display it on PC.

Desired Implementation:

1. The ADC data should be properly stored in the buffer provided.
2. The data should be filtered properly so that noise does not affect the working of system.
3. The firebird should move in the desired direction upon sensing the gunshot.

Actual Implementation:

1. ADC data is properly stored in the buffer of DMA RAM.
2. Filtering is done by circular convolution. Since we did not test the system with a real gun, the toy gun used had frequency components with a center frequency of around 5Khz. Since humans have a audio components upto 5 Khz a very loud shouting near the sensors can affect the system. This will not cause a problem with a real gun as the frequency signature in this case cannot be replicated in nature.
3. Since the testing was done in closed space and sensitivity of each sensor being different sometimes a different sensor picks up the sound rather than the one actually desired. So the firebird sometimes moves in wrong direction.

7. Future Work

1. The filtering can be done better. We used as 12MHz external oscillator for our dsPIC. So the internal clock frequency used is 6 MHz. This severely limits the rate at which ADC can capture samples. The frequency can be increased in dsPIC by using higher external oscillator or using internal oscillator functionality of dsPIC.
2. Current sensors can be replaced by sound pressure level meters which are much more accurate.
3. Image processing can be included in the system using dedicated DSP processor.

8. Conclusion

We would like to conclude by saying that this system has wide ranging application in field of homeland security. Many more features can be added to this system to make it more robust and faster. The code written is well commented and can be readily reused. The logic used is simple and can be applied for any controller. We learnt a lot of concepts related to embedded controller programming, issues arising during design and testing and had a lot of fun working on new controllers. On the non-technical side, we learnt how to go about designing the schedule and managing the time and also inter personal communication. We would like to thank Prof Kavi Arya along with the TA and lab staff for their support during the course of this project.

9 References

[1] dsPIC33FJ128GP802 datasheet
[2] dsPIC family reference manuals
[3] Microchip support website www.microchip.com
[4] Firebird Software Manual