ECE 7220

Real Time Embedded Systems

Final Project

Speeding Detecting System

By

Hancheng Wu

Abstract

Speeding is one of the most common reasons that lead to traffic accidents. This project implements a prototype of speeding detecting system used to detect speeding behavior on the road. This embedded system has a good real time performance due to the use of interrupts. Project is based on TS-7250 board.

Keywords

REAL TIME, SPEEDING DETECTIING, EMBEDDED SYSTEM

Introduction

Report shows that more than sixty percent of the traffic accidents are caused by speeding. Many devices are used to help police detect the speeding behavior.

A prototype of speeding detecting system is proposed in this project. The design uses interrupts to calculate the speed of passing vehicles. Once speeding happens, the picture of the speeding car will be captured and stored for future reference.

Due to the use of interrupts and periodic tasks, the system has a good real time performance.

All the design is based on TS-7250 board. There are totally five software blocks running on four boards. Some simulating modules are built for experiments.

Background

There are various systems used to detect speeding behavior in real world. Three of them are commonly seen.

Calculate speed via camera

The most convenient way to achieve this target is detect the speed via camera. A special image processing program could be embedded into the camera. The camera keeps videotaping the road and will recognize the vehicles on the road. Via some algorithm it could detect the moving of the vehicle then it could get the speed of the passing vehicle by dividing the distance by the time the movement uses. This implementation has an obvious advantage which is that it is easy to be built in real world. All needed is a camera with an embedded image processing program. Disadvantage of this system is that it is not accurate and when there are too many cars it might miss some cars because of the overloading of the image processing program.

Speeding detecting with radar

Another implementation is detecting the speed via radar. No need to say this speed detecting method is going to be as accurate as possible. Once speeding happens, radar will send a signal to the camera so that a picture of the speeding car will be captured. This implementation has a good real time response and is not complicated to build for real world projects. The disadvantage of this approach is that it is expensive because of the use of radar.

Speeding detecting using sensors on the road

Other than the two methods mentioned above, sensors are often used to detect the speed. Two sensors are set on the road with a certain distance. A passing vehicle will trigger the sensors respectively, therefore two signals will be sent to a module. Speed could be acquired given the interval time of the signals and the distance of the sensors. This way is commonly used because it is accurate and the cost is affordable. Disadvantage is that reconstruction of the road has to be made to put the sensors in.

Method of this project

Method used in this final project is based on the previous method. Usually a speed detecting module is a hardware module separately distributed with main device, and it is not changeable. Difference of this project is that speed detecting module and image acquire module are combined together. They are running on the same TS-7250 board and can be easily changed simply by modifying the program. Interrupts are used to ensure the accuracy of the speed detecting.

Proposed method

System structure

The system consists of five blocks of software running on four boards. See Figure 1.

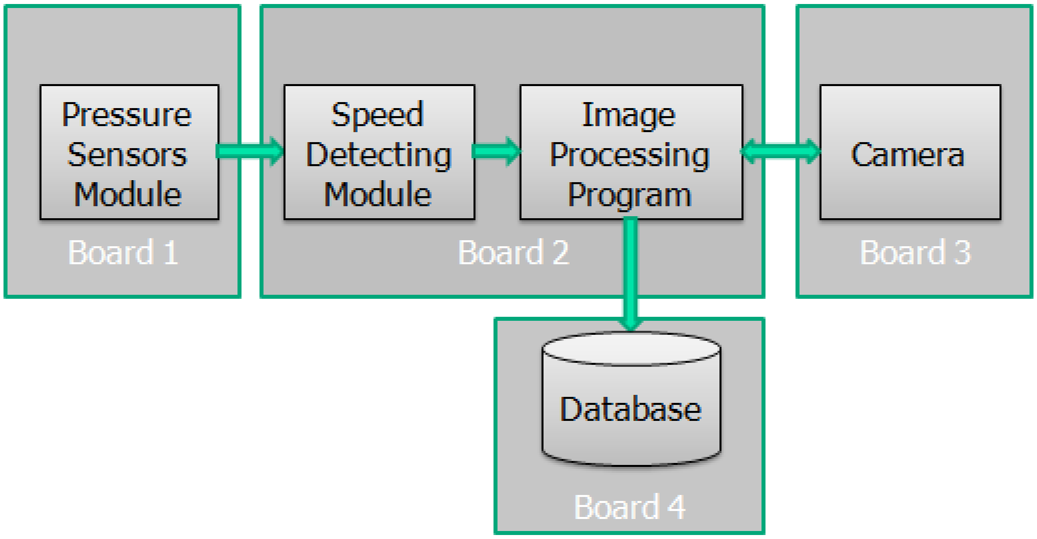
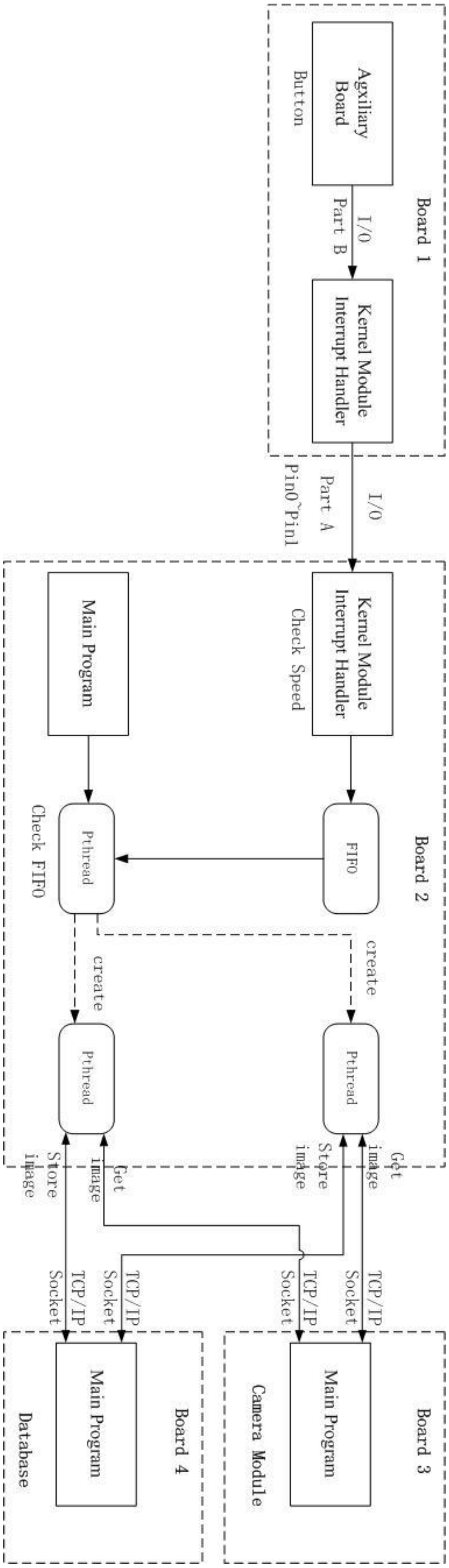


Figure 1

Board1 and board2 are connected via I/O portA. Board1 will send signals through pin0 and pin1. Board2 will take these two signals as interrupts. Board2 are communicating with board3 and board4 via local network.

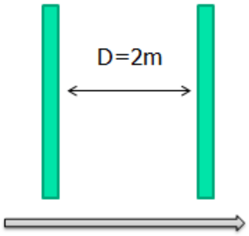
Pressure sensors module runs on board1. Speed detecting module and Image processing program run on board2. Camera module and database module run on board3 and board4 respectively.

See the program structure below.


Board 1
Agxiliary Board
Button
I/O
Part B
Kernel Module Interrupt Handler
I/O
Part A
Pin0~Pin1
Board 2
Kernel Module Interrupt Handler
Check Speed
FIFO
Main Program
Pthread
Check FIFO
create
create
Pthread
Pthread
Get image
Store image
Get image
Store image
TCP/IP Socket
TCP/IP Socket
TCP/IP Socket
TCP/IP Socket
Board 3
Main Program
Camera Module
Board 4
Main Program
Database

Pressure Sensors Module

Pressure sensors module simulate two pressure sensors which will generate two signals with uncertain interval time.



$S=D/(T2-T1)$

See above picture. Suppose the distance between two sensors is 2 meters. Then T2-T1 should be constrained by Distance/Speed. If this module is going to generate signals which are representing speed ranging from 35miles/h to 120miles/h. Then the T2-T1 should be in the range of 2m/(35miles/h) to 2m/(120miles/h), which is from 37ms to 128ms. Thus, the key problem in this module is to generate signals with interval time ranging from 37ms to 128ms.

Algorithm used

When interrupts happens, do the following.

1) Use gettimeofday() function to get the current time
2) Use the last 7 bits of the time as the random number
3) Linear transfer the number to 37 to 128, let it be number2
4) Post signal1, which means vehicle passes the first sensor
5) Wait for number2 ms
6) Post signal2, which mean vehicles passed the second sensor

By this way, time constraints will be met.

Implementation

PortB are set as interrupt source.

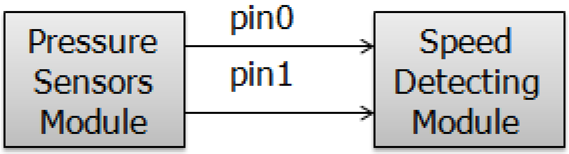
When buttons on auxiliary board are pressed, interrupt handler is called.

Above mentioned algorithm is implemented in the interrupt handler.

Pin0 and pin1 of portA are used to send the two signals respectively.

Speed Detecting Module

This is a kernel module with interrupt handler running on board2. The interrupt handler will detect if there is speed happening.
This kernel module set pin0 and pin1 of portA as interrupt source. Interrupt is set as falling edge triggered. See figure below.



This module calculates the speed of the car according to the signals on portA.

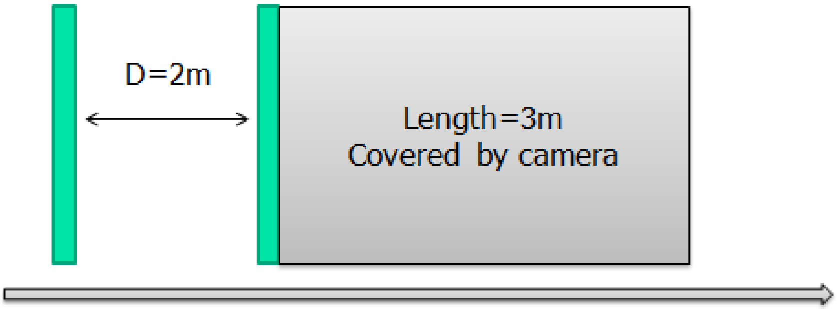
Usually a speed of over 75 miles per hour will be considered speeding, which means that a case with a signal interval time less than 59.6ms is a speeding.

```
Implementation
When port0 triggers the interrupt, record the system time as timestamp1;
When port1 triggers the interrupt, record the system time as timestamp2,
    then check if (timestamp2- timestamp1)<59.6ms.
    If true, then speeding occurs. The speed will be written into a fifo to
    notify a user space program.
Exit the interrupt handler.
```

Image Processing Program

This is a user space program running on board2. The program will receive FIFO signal from kernel module interrupt handler and then require image from camera module, after the image is received, it will store the image on the database.

Real time constraint



As showed above, suppose the camera can cover 3meters long after the sensors. Real time constraints must be applied to the design so that the camera will triggered in time. Otherwise the vehicle will be out of the covered area.

Suppose the worst case, where the car meets a speed of 120 miles per hour, which is 53.648 meters per second. As a result, the car will leave the covered section in 55.92 ms.
In order to capture the image of the speeding car, the camera has to be triggered within the time of 55.92 ms. Be more strict, cut the time to <u>20 ms.</u>
In the design, the program will read the fifo signal with a period of 4ms. It turns out this will perfectly meet the real time requirement.

Implementation

1) Main program creates the check_speed pthread.
2) The check_speed pthread periodically read the FIFO from the kernel interrupt. Whenever a signal is received, another pthread get_store_image is created. The check_speed pthread will go on check the fifo.
3) The get_store_image pthread requires image from camera module and store the image to the database. After that, it will exit.

Camera Module

Camera module behaves like a camera. It is used to replace a camera because TS-7250 does not support camera.
It is actually a server running on the third board. Upon request, it will send back an image.

Database

Database program is a server running on the fourth board. Upon request, it will receive the image and store it.

Results

In all the tests, the system meets the real time requirement and acquires and stores the image of the speeding car successfully.
Some results are showed below.

```
car speed: 51 m/s,  signal time: 2214502210 us,  current time: 2214509086 us
Response time: 6876 us
Speeding! Camera triggered!
Receiving image! Image plate0.bmp created!
Image received!
From receiving signal to receiving piture:
218861 us
Picture transferred!

car speed: 36 m/s,  signal time: 2217056918 us,  current time: 2217063530 us
Response time: 6612 us
Speeding! Camera triggered!
Receiving image! Image plate1.bmp created!
Image received!
From receiving signal to receiving piture:
195720 us
Picture transferred!

car speed: 51 m/s,  signal time: 2220510034 us,  current time: 2220516598 us
Response time: 6564 us
Speeding! Camera triggered!
Receiving image! Image plate2.bmp created!
Image received!
From receiving signal to receiving piture:
204370 us
Picture transferred!

car speed: 36 m/s,  signal time: 2222595974 us,  current time: 2222600765 us
Response time: 4791 us
SpeedingCamera triggered!
Receiving image! Image plate3.bmp created!
Image received!
From receiving signal to receiving piture:
228485 us
Picture transferred!
```

```
car speed: 42 m/s,  signal time: 2232549101 us,  current time: 2232553948 us
Response time: 4847 us
Speeding! The camera will be triggered immediately!
Camera triggered!
Receiving image! Image plate4.bmp created!
Image received!
From receiving signal to receiving piture:
228882 us
Picture transferred!
```

Car speed is showed. If a speed of larger than 75 miles is considered speeding, which means that a speed of larger than 33.53m/s will be considered speeding.
Signal time is the time when speeding signal is received.
Current time is the time then camera is triggered.
Response time is the time from signal time to current time.
From receiving signal to receiving piture is the time taken to store the picture.

From above results, it can be seen that the response times in the result are respectively 6.876ms, 6.612ms, 6.564ms, 4.791ms and 4.847 ms.

They are much less than the real time constraint discussed in the previous section, which is 20 ms.

Real time constraint is perfectly met in this design.

How image is acquired from camera module and stored to database is showed in the attached video.

Conclusions

As final project of embedded system course, many methods learned from course are implemented in this project. This project meets the real time constraint and achieves its design requirements.

However, some improvements might be able to made to improve the design.

First, now the speeding detecting system is a one-camera-one-system design. Can it be changed to multiple-lanes-one-system? That mean the system can take care of multiple lanes and multiple cameras. Of course, real time constraints are still needed to be met. As a result, CPU speed is a main issue.

Second, some image recognition program can be applied on the database. So after the image is stored, it will be automatically analyzed. License plate will be recognized and stored.