Abstract

This paper describes a real-time online prototype driver-fatigue monitor. It is about making cars more intelligent and interactive which may notify or resist user under unacceptable conditions, they may provide critical information of real time situations to rescue or police or owner himself. It uses remotely located charge-coupled-device cameras equipped with active infrared illuminators to acquire video images of the driver. Technological approaches for detecting and monitoring fatigue levels of driver fatigue continue to emerge and many are now in the development, validation testing, or early implementation stages. The main objective of this document is description of the control and monitoring system with integrated image processing from the camera. The image obtained from the camera is used for recognizing routing and traffic situation. The main components of the system consist of number of real time sensors like gas, eye blink, alcohol, fuel, impact sensors and a software interface with GPS and Google Maps APIs for location.
1. Introduction

The causes of vehicle accident in recent years are due to inadequate driving. If driver is drowsy or alcoholic accident causes. Driver drowsiness is recognized as an important factor in the vehicle accidents. It was demonstrated that driving performance deteriorates with increased drowsiness with resulting crashes constituting more than 30% of all vehicle accidents. But the life lost once cannot be re-winded. Using advanced technology the accidents are reduced to some extent. This project involves controlling accident due to Eye blinking. Here one eye blink sensor is fixed in vehicle. An alarm is fixed to indicate unconscious blinking. Here the project involves physiological data for validation of this technology. Methodology for analysis of physiological data, independent assessment of driver drowsiness and development of drowsiness detection algorithm by means of sequential fitting and selection of regression models is presented.

We can’t take care of ours while in running by less conscious. If we done all the vehicles with automated security system that provides high security to driver, also gives alarm. we can incorporate it with a special instruction written in image processing that, if there is no pupil found for the certain period of pre-determined i.e. time greater than the human eye blinking time then consider an event called “blink”, for which the set of operations will be followed. Here, in this case we need to set time as 1 second or above it, as “blink event” is different from “normal eye blinking”. We need to perform testing for only blink event estimation, and not to find normal eye blinking.

2. Literature Survey

S.P. Bhumkar, V.V. Deotare, R. V. Babar investigated in this paper “Accident avoidance and detection on highways” is when you think of work-related safety hazards, you probably think about what goes on inside the workplace. Boneo the greatest threats to your safety are not in the workplace, but rather on the road. Someone is injured every 18seconds. Over 2million of those injuries turn out to be disabling. A person dies in a crash on U.S. roads every 11 minutes. In fact, motor vehicle accidents are the most common cause of death in the United States more than cancer or heart attacks. When we think about the serious accident, it could change your life- and not for the better [1].
K. Srijayathi, M. Vedachary investigated in, “Implementation of the Driver Drowsiness Detection System” is Driver fatigue is a significant factor in a large number of vehicle accidents. The development of technologies for detecting or preventing drowsiness at the wheel is a major challenge in the field of accident avoidance systems. Because of the hazard that Drowsiness presents on the road, methods need to be developed for counteracting its affects. The aim of this project is to develop a prototype drowsiness detection system. The focus will be placed on designing a system that will accurately monitor the open or closed state of the driver’s eyes in real-time. In today’s world where science has made amazing advances so have the recent cars. These cars are more advanced than ever. But now a day, due to driver drowsiness accidents are increasing day by day. Driver Drowsiness and then they do rash driving as of that they do not have control on themselves. Here we designed a system which will detect driver drowsiness.3.

Abhi R. Varma, Seema V. Arote, Chetna Bharti investigated in, “Accident Prevention Using Eye Blinking and Head Movement” is that describes a real-time online prototype driver-fatigue monitor. It uses remotely located charge-coupled-device cameras equipped with active infrared illuminators to acquire video images of the driver. Various visual cues that typically characterize the level of alertness of a person are extracted in real time and systematically combined to infer the fatigue level of the driver. The visual cues employed characterize eyelid movement, gaze movement, head movement, and facial expression. A probabilistic model is developed to model human fatigue and to predict fatigue based on the visual cues obtained.4.

S.P. Bhumkar, V.V. Deotare, R.V.Babar investigated in, “Intelligent Car System for Accident Prevention Using ARM-7” that is about making cars more intelligent and interactive which may notify or resist user under unacceptable conditions, they may provide critical information of real time situations to rescue or police or owner himself. Driver fatigue resulting from sleep deprivation or sleep disorders is an important factor in the increasing number of accidents on today’s roads. In this paper, we describe a real-time online safety prototype that controls the vehicle speed under driver fatigue.2.

3. System Overview

A. Beagle Board

Beagle Board promotes and motivates open source development on ARM® microprocessor-based systems.

The key objectives of the organizations are to:
• Enable hobbyists and innovators to explore new domains and experiment with their ideas on an open platform
• Enable such experiments to be conducted cost effectively − to nurture innovation − by enabling the supply of ultra-low-cost hardware platforms
• Bring together communities by providing the basic infrastructure to exchange ideas

This document is the System reference Manual for the Beagle Bone Black. It covers the design for the Beagle Bone Black. The board will be referred to in the remainder of this document as Beagle Bone Black. There are also references to the original Beagle Bone as well. This design is subject to change without notice as we will work to keep improving the design as the product matures.

B. Processor
OMAP3530 Microprocessor − 1200 DMIPS, based on ARM Cortex-A8 running at 600Mhz
• TMS320C64x+ DSP for versatile signal processing at up to 430MHz
• Memories: 256MB of NAND Flash, 256MB of SDRAM
• SD/MMC Card slot, can be used as OS file system, file storage and more
• USB Host port which gives the option to connect a full set of peripherals using a hub (Ethernet adaptor, keyboard, mouse...)
• DVI-D digital output and S-VIDEO (NTSC/PAL) output
• Stereo audio output and microphone input
• Multiplex able expansion header: I2C, SPI, UART, GPIO, SD/MMC
• Typical power consumption: 2W (at 5V)

All those features have to managed by an operating system in order to use them in an efficient way. The best option for BB is Linux, mainly because there is an active project supporting BB which includes working drivers for the features listed above, a good tool chain (GCC based) and many common packages ready to build and install into BB. In this document you will find the steps to get a fully functional Linux running on your BB and an introduction on how to put your own software into BB.

C. Linux On Beagleboard
Linux alone is no more than an operating system kernel, which is an essential piece but not enough to get a fully functional system. You usually will need a shell and a basic toolbox giving you access to functions such as browse directories, read and write files, etc. After that many extra packages can be installed on your file system, even your own software.

The Angstrom distribution and Open Embedded

Angstroms complete Linux distribution: includes the kernel, a base file system, basic tools and even a package manager to install software from a repository. It is optimized for low-power microcontrollers like the one in BB and intends to be small and basic system to modify on your needs. It uses the Open Embedded (OE from now on) platform, a tool chain that makes cross compiling and deploying packages easy for embedded platforms. There are other options available; however this document is focused on Angstrom.

Step 1: Get Open Embedded metadata
First you will need to get the OE metadata from their GIT repository (information, recipes, tools, etc.). Remember that you need to have GIT installed before typing the following commands.
## Step 2: Configure Open Embedded

When you have downloaded all metadata files, you need to set up OE indicating your target board, directory of OE recipes, etc. While inside OE root folder, create a new folder tree:

```
$ mkdir -p build/conf
```

Inside `conf` directory, create a file named `local.conf` and write these lines:

- `BBFILES = "/path/you/want/for/OE/open embedded/recipes/*/*.bb"`
- `DISTRO = "angstrom-2008.1"
- `MACHINE = "beagle board"

# Use N threads (write only if you have a multicore machine)
# Change N for the number of your processor cores
`BB_NUMBER_THREADS = "N"
`PARALLEL_MAKE = "-j N"

Finally, on your OE root folder, create a file named `profile` with these lines:

- `export OE_HOME=/path/you/want/for/OE`
- `export BBPATH=${OE_HOME}/build:${OE_HOME}/open embedded`
- `export PATH=${OE_HOME}/open embedded/bitbake/bin:$PATH`

The `profile` is a file that is need to be run every time you want to start using OE environment. It basically sets the working directories so OE tools can find all files.

## Step 3: Build software using Open Embedded

Now we are ready to start building software for your BB using OE configured to use Angstrom distribution. OE uses `bitbakeas` a package builder and manager. Using the previously downloaded recipes, a set of instructions to build a package, it will make everything for you when you need to build a package, even solve dependencies. When you run `bitbake` for first time it caches all recipes and it downloads a cross-tool chain so it has a cross-compiler to build other packages. The first thing you should build is a base operating system for your BB. In the case of Angstrom, there are 3 options:

- **Base Image**: A minimal operating system
- **Introduction to Beagle Board**
- **Console Image**: A basic operating system including some extra tools (ex. networking)
- **X11 Image**: An operating system including graphical interface

You can choose whatever you want depending on your needs.

First, load the profile (remember to do that every time you start working with OE):

```
$ source profile
```

Now, you can build Angstrom by typing:

```
$ bitbake -f base-image
```
$ bitbake -f console-image
$ bitbake -f x11-image

Build process will take a long time (about 6 hour for console-image) as it needs to download many packages from the net and build them.

All files involved in build process (sources, logs, object files...) will be placed in “tmp” folder.

4. Implementation

![Block diagram of Hardware Implementation](image)

In above block diagram it is shown that various sensors that we are going to use are interfaced with the beagle board. The beagle board is actually, Here sensor n alarm platform which has all the sensor and a speaker to be mounted on the driver LCD unit for visual interface to the system amplification n filter unit for sending enlarge alarm sound to the speaker sensor and framework.

Power supply for powering all the Modules we are going to use the above sensors:

**Gas Sensor: MQ6**
High sensitivity to propane, butane and LPG also responsible for natural gas

Application
- Domestic gas leakage detector
- Industrial combustible gas detector
- Portable gas detector

**Alcohol Sensor: MQ3**

Applications
- Vehicle alcohol detector
- Portable alcohol detector

**Accelerometer Sensor**: MMA7760accelerometer and an Infra-Red Sensor Breakout Board consisting of an LM358 Op Amp by Texas Instruments and a potentiometer for calibration.

**Temperature Sensor**: LM35, LM34

**Parking proximity sensor**
PROXEL ESP dual 2.0 complete rear parking sensor
- Continuous sound low frequency (risk of contact) the bumper is 15-10 cm close to obstacle.
- Continuous sound at high frequency ESP informs that the obstacle is at a distance variable between 30 n 15 cm from the bumper.
- Bits with increasing frequency (alert signal) ESP alerts you of presence and approach of obstacle.

5. Image Processing
Modern digital technology has made it possible to manipulate multi-dimensional signals with systems that range from simple digital circuits to advanced parallel computers. The goal of this manipulation can be divided into three categories:
• Image Processing image in → image out
• Image Analysis image in → measurements out
• Image Understanding image in → high-level description out

We will focus on the fundamental concepts of image processing. Space does not permit us to make more than a few introductory remarks about image analysis. We begin with certain basic definitions. An image defined in the “real world” is considered to be a function of two real variables, for example, a(x,y) with a as the amplitude (e.g. brightness) of the image at the real coordinate position (x,y). In a sophisticated image processing system it should be possible to apply specific image processing operations to selected regions. Thus one part of an image (region) might be processed to suppress motion blur while another part might be processed to improve colour rendition. The 2D continuous image a(x,y) is divided into N rows and M columns. The intersection of a row and a column is termed a pixel. The value assigned to the integer coordinates \([m,n]\) with \(\{m=0,1,2,...,M-1\}\) and \(\{n=0,1,2,...,N-1\}\) is a \(a[m,n]\). In fact, in most cases a(x,y) – which we might consider to be the physical signal that impinges on the face of a 2D sensor – is actually a function of many variables including depth (z), colour (\(\lambda\)) and time (t).

Figure 5.1: Image Detection in Car

6. Future Scope
This system only looks at the number of consecutive frames where the eyes are closed; at the point it may be too late to issue the warning. By studying eye movement patterns, it is possible to find a method to generate the warning sooner. The purposed system is used to avoid various road
accidents caused by drowsy driving. Instead of alarm we can use Automatic Braking System which will reduce the speed of the car.

- We can automatically park the car by first using Automatic braking system, which will slow down the car and simultaneously will turn on the parking lights of the car and will detect the parking space and will automatically park the car preventing from accident.
- Using Pressure sensor on the steering alarm or Automatic braking System can be set in case of drowsiness.
- By using wire-less technology such as Car Talk2000 If the driver gets a heart attack or he is drunk it will send signals to vehicles nearby about this so driver become alert.

7. Conclusion

It is due to the driver’s fatigue, traffic accidents keep with a yearly increasing of a high rate. This shows the new fatigue detection algorithm & techniques using eye blink, alcohol, impact, gas, etc. sensors. In this technique the fatigue will be detected immediately and regular traps the events driver and the third party. We propose an intelligent car system for accident prevention and making the world a much better and safe place to live.

References


Website:
[1] 1.eLinux.org/BeagleBoard

Books: