ON-BOARD COST EFFECTIVE RECORDING SYSTEM

FOR ENHANCED ROAD SAFETY

Naji Majdalani

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfilment of the Requirements

for the Degree of Master of Applied Science (Mechanical Engineering)

at Concordia University

Montreal, Quebec, Canada

July 2003

National Library of Canada

Acquisitions and Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada

Acquisisitons et services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-83884-6
Our file Notre référence
ISBN: 0-612-83884-6

ABSTRACT

ON-BOARD COST EFFECTIVE RECORDING SYSTEM FOR ENHANCED ROAD SAFETY

Naji Majdalani

Road safety enhancement requires effort from both parties involved: the users and the enforcing authority. While the costs on maintaining and enhancing the safety of the roads is growing, statistics show that vehicle related accidents are still quite high. The presence of the high tech in vehicle has not altered too much the gravity and even the number of accidents. Moreover, accidents recording are often contradicting or fuzzy since significant information is not brought to the attention of the accident recording authority. A clear and distinct data collection would put the bases of the awareness of specific maneuvers or driving habits that would be associated with a high risk of accident. Onboard recorders are presently making their way into the automotive industry and especially in the commercial vehicles. The heavy vehicle represent a high risk when involved in accidents due to their high inertia and reduced lateral and longitudinal stability. The many types of recorders have been implemented to monitor drivers and their driving behavior, but they are considered by the drivers unions as "invasion of privacy" tools. However, the main drawback is the fact that they are expensive to drivers. Such systems have proved extremely useful in evaluating the cause of accidents and the associated liability. The truck drivers using it do not perceive it as a privacy invasion tool.

A low cost monitoring system based on Smart Card technology is designed and developed to monitor the entire duration of driving, duration of pause, average driving

speed as well as peak speeds over a preset period of time. The system is recording the information on a 8KB smart card of an encrypted manner to avoid any interaction of the driver or the enforcing authority with the information. The encryption permits continuous records as long as 30 days and could accommodate the information collected from the on-board computer of the car as well as of the external sensors. The system is using a microcontroller with serial and parallel interfaces to collect data. The system has been conceived as expandable to a larger number of sensors as well as to a larger capacity smart card system. In addition, an investigation on the capability of the system to record other information related to the driver's driving habits or dangerous manoeuvres has been carried out. The system has also been conceived to be capable of reading information from remote sensors through telemetry and record vehicle conditions that may lead to future accidents.

ACKNOWLEDGMENTS

The author wishes to express his sincere appreciation to his thesis supervisors, Dr. I. Stiharu and Dr. H. Hong for initiating the study topic and providing guidance throughout the course of this investigation.

Thanks are due to the faculty members, staff and other graduate students of CONCAVE Research Center, and the Department of Mechanical and Industrial Engineering for their help during the course of this work.

The financial support provided by MTQ-SAAQ is gratefully appreciated.

Finally, the author would like to express his special thanks to his family members for their love, encouragement and support.

TABLE OF CONTENTS

PAGE

LIST OF FIGURES

Figure		PAGE

Figure		PAGE

LIST OF TABLES

NOMENCLATURE

SYMBOL	DESCRIPTION
a	Trajectory parameter
A	Longitudinal acceleration
A_1	Leading vehicle longitudinal acceleration
A_2	Following vehicle longitudinal acceleration
A_{lat}	Lateral acceleration
APDU	Application Protocol Data Unit
ATR	Answer to reset
b	Trajectory parameter
c	Trajectory parameter
C	Capacitor
CLA	Class byte
CLK	Clock input
COS	Card operating system
CPU	Central Processing Unit
CTS	Clear to send
d	Trajectory parameter
D	Adjustment factor
DB-9	9 Pin connector
DB-25	25 Pin connector
DC	Direct current

DTR	Reset input signal
EEPROM	Electrically erasable read only memory
etu	Elementary time unit
ETX	End of transmission byte
f	Operating frequency
F	The clock rate conversion factor
F1	Load reaction on right wheel
F2	Load reaction on left wheel
F_{osc}	Frequency of oscillation
FSR	File select register
GND	Ground
GPR	General Purpose Registers
H	Height of the center of gravity
HS	High Speed
I/O	Input/output
INS	Instruction byte
Lc	Length of the data section
Le	Length of expected response
MCU	Microcontroller unit
N	Number of User Data Files defined
OSC1, OSC2	Frequency oscillator
P1, P2	Parameter bytes
PVC	Polyvinyl chloride

r	Resistor
R	Radius of curvature
RAM	Random access memory
RE0, RE1, RE2	Port E pins
ROM	Read only memory
RS-232	Serial Communication protocol
RST	Reset Input
RxD	Receive data
SD	Safe braking distance
SFR	Special Function Registers
STX	Start of transmission byte
SW1, SW2	Response status
t	Time
T	Distance for the center of the vehicle to a tire
Ti	The historical character
T0	The format character
TA1, TB1, TC1, TD1	The global interface characters
TAi, TBi, TCi, TDi	The interface characters
TCK	The check character
TS	The initial character
TxD	Transmit data
UART	Universal Asynchronous Receiver Transmitter
V_i	Initial vehicle velocity

V_f	Final vehicle velocity
V_1	Leading vehicle velocity
V_2	Following vehicle velocity
Vcc	Supply voltage
W	Working register
W	Truck Weight
X	8-bit value
XOR	Xoring logic
y	Trajectory equation
$\frac{dy}{dx}$	First derivative of trajectory
$\frac{d^2y}{dx^2}$	Second derivative of trajectory
Äx	Distance travelled by vehicle
Äy	Distance travelled by the c.g while turning
ô	Output timing

CHAPTER 1

LITERATURE REVIEW AND SCOPE OF THE DISSERTATION

1.1 Introduction

There are over 600 thousand trucks operating in all Canada and this number is increasing to cope with the growing economy [1]. Low cost in freight transportation as well as being the only land transportation to supplement the railroad infrastructure, have helped this field flourish and become an economical pillar. Unfortunately, the economical dependence on road transportation and the huge profits truck companies are making is pushing commercial vehicle drivers to the limits. Drivers are expected to transport freight while providing on-time deliveries, undamaged product and customer satisfaction [2]. As a result, every year truck accidents kill over 400 people and injure almost 40,000 more. A study of highway safety in the U.S. [1] concluded that in 1998, 98% of the fatalities in two-vehicle crashes involving passenger vehicles and large trucks were occupants of the passenger vehicles [1]. In recent years almost one in every four fatalities among passenger vehicle occupants has been the result of multi-vehicle collisions involving a large truck. A more recent study of heavy truck collisions [3] claimed that during the period extending from 1994 to 1998, an average of 43,843 accidents involving heavy trucks occurred each year in Canada resulting in an average of 11% fatal collisions.

The need for rules and regulations has become imminent in order to reduce accidents that not only claim human lives but are also an economical burden for governments. According to Grace et al. [4], there are 1.6 million truck tractors and 3.6 million trailers used in the motor carrier industry in the U.S., which account, yearly, for

1,200 deaths and 76,000 injuries related to fatigue and drowsiness. The cost of these crashes is estimated at $12.4 billion per year. In the European Union, according to a survey conducted by the European Land Transport Agency [5], emergency services and medical care of fatal accidents cost a total of € 45 billion a year, or € 1 million for every person killed. The toll motivates efforts to identify and implement effective countermeasures like the installation of tachographs on trucks to measure and monitor speed, fatigue, drowsiness, driving time as well as several other parameters. Crash Communicator [6] stated that new U.S. rules require long-haul trucks to be equipped with electronic recorders to stop reported substantial violations of limits on hours whereas Canada is proposing to continue with the discredited logbook system.

Much progress has been made over recent years in understanding many aspects of driver-vehicle interactions, mainly interactions with Intelligent Transportation Systems (ITS). However the commercial approach to the design and development of any aspects of the vehicle, which influences its dynamic properties is based on a coordinated approach using three aspects: modeling, experiments and subjective assessments [7]. The solution proposed in this dissertation is based on a low cost monitoring system capable of measuring adequate and suitable data while still being affordable. It aims at monitoring truck drivers who are flirting with safety in order to make roads safer and prevent fatal accidents. Data generated by a C++ code, which simulates the onboard truck computer, is sent to a microcontroller through parallel communication. The data is processed, averaged and sent to a smart card reader, which stores information on the Smart Card. The Smart Card is a credit card sized plastic card embedded with an integrated circuit chip [8]. This electronic system is designed to record truck drivers driving habits and store the collected data in a safe and secure memory device, which could be safely accessed in critical situations in order to better understand crash causes and determine drivers responsibilities. A brief review of the literature on accident statistics related to speed, fatigue and drowsiness as well as previous implemented solutions and an introduction to Smart Cards are presented in the following sections to formulate the scope of this dissertation research.

1.2 Problem Overview

Transportation has always been the main issue in traveling, shipping, delivering and transporting people and goods. Air, water and land transportation have flourished during the past century and have become a crucial part of our way of life. Vehicles have become an indispensable tool and a driver for the economy. Companies rely on commercial vehicles to send their merchandise all over the land, cars and trucks have become faster, more reliable, more intelligent and human friendlier, companies have also realized the importance of transportation in their economical blooming. All these effects have helped increase the number of vehicles on the roads and in particular heavy vehicles, but this development in road transportation has not been all beneficial. The increase in number of vehicles has led to higher percentage of accidents, fatal in many cases, and made people realize that safety issues are now at stake. Furthermore, truck drivers have been violating the truck driving code by driving longer hours than permitted and over speeding as well as committing logbook infractions. Many solutions have been proposed or implemented over the past few years to overcome such violations and enhance road safety.

1.2.1 Road Safety

The continuous increase of vehicles on the roads as well as transportation safety issues have become a major concern on government agendas due to the increasing percentage of vehicle accidents. In Canada alone, in the year 2000, having 18 million vehicles registered for 20 million licensed drivers, traffic accidents claimed the lives of over 3,000 road users and injured another 220,000, many of them seriously. Among those, only 8% were accidents involving commercial vehicles but account for 18% of the fatalities due to the large vehicle mass [9]. Numbers cited above motivate people and push governments to act fast to set rules and regulations in order to insure safer roads specially when it comes to heavy vehicles, which have an additional set of rules to abide by.

1.2.2 Commercial Vehicle Economical Impact

People encounter trucking in their daily lives in one way or another. Trucks represent a safe, economical and in many cases "the only way" of transporting goods across the land. Small and big companies rely on them to get or send their merchandise. In Canada for example, trucking accounts for significant revenues and jobs across the country. Estimates suggest trucking is a $31 billion industry and the for-hire sector accounts for nearly half of it. For-hire trucking activities produce approximately 158,000 jobs. Virtually every product a consumer purchases has been transported by truck at least part of the way, sometimes several times before reaching a final destination [9]. Approximately 118,000 large trucks in Canada haul freight commercially in for-hire operations. Non-commercial trucking operations use farm, utility and service trucks. The courier business that transports mail and small packages also uses trucks. Trucking has become very popular and very important to the economical survival of businesses because it offers a flexible mode of transport, constrained only by the extent of the road network and due to this flexibility, trucking has been able to provide the kind of services required by even the most demanding shippers.

1.2.3 Traffic Control of Heavy Vehicles

Needing to compromise between the importance of trucks in the economical survival of businesses and regions, as well as the danger they represent on the roads, governments have set rules and regulations to control commercial vehicles [18]. Among the conditions that were stressed upon by authorities was the speed limit, driving hours that result in driver fatigue and drowsiness as well as drinking. In addition, trucks have been equipped with several monitoring devices and state of the art communication systems to help drivers control and manage themselves along with their trucks.

1.2.4 Road Accidents Involving Heavy Vehicles

The department of transportation in Great Britain [10] published a fact sheet in 1999 that was later updated in 2002 stating that in 1998, there was 18,698 casualties in Heavy Goods Vehicles (HGV) accidents, an increase of 5% on the baseline average of 17,808 but on the other hand, deaths and serious injuries have both passed the target of a one third reduction in the same period. Of the 576 killed in those types of accidents, just 60 were occupants of the HGV; a fifth of the fatalities were cyclists and pedestrians. Figure 1.1 shows that accidents involving HGVs has decrease over the last two decades by about 50%, but the numbers still remain high. In 1998, there were around 15,000 road accident casualties, and among those were 4,000 killed or seriously injured.

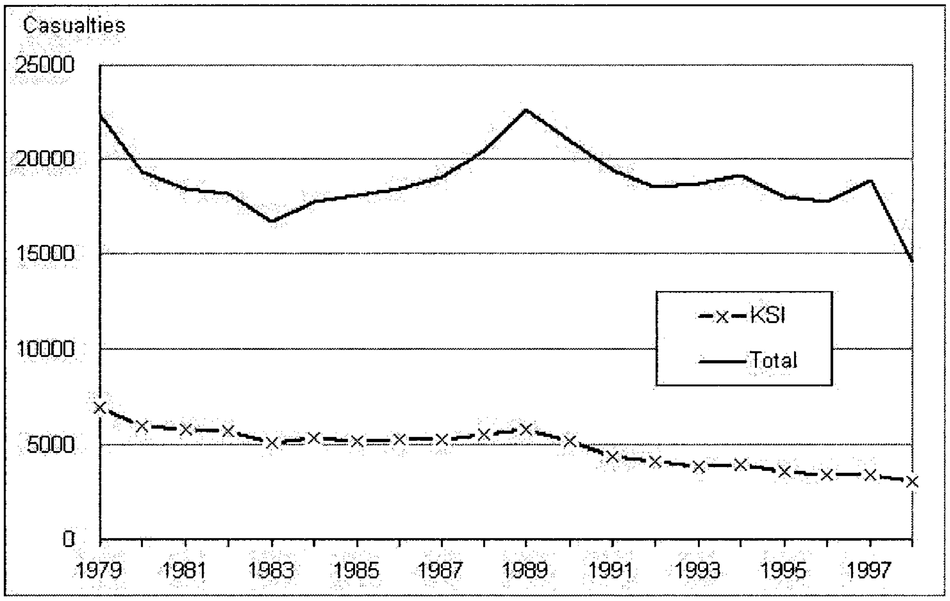


Figure 1.1: Casualties killed or seriously injured (KSI) in accidents involving HGVs in Great Britain :1979-1998 [10].

Michel et al. [11] reported that nearly 75% of all accidents involving a heavy vehicle and a passenger vehicle are attributed to a human error. The report also states that heavy vehicle-related accidents involving bodily injuries remain at about 2,200 a year throughout the 1990s, despite the means implemented under conventional road safety programs. Transport Canada [3] reports that straight trucks were involved in an average of 8,169 collisions resulting in 5.2% per year of all injury-producing collisions that occurred from 1994 to 1998. In addition, the report states that the greatest number of fatal and personal injury collisions involving commercial vehicles occurred on undivided highways, roads and streets. Seventy-nine percent of fatal collisions involving straight trucks occurred on undivided roadways whereas 15% happened on divided highways.

Figure 1.2 shows the percentage distribution of collisions involving heavy trucks by collision configuration and severity.

Collision Configuration	Fatal	Personal Injury	Property Damage
2 M.V.: Head-On (& Sideswipes)	35.2	7.3	3.7
2 M.V.: Right Turn	14.0	12.3	6.9
1 M.V.: Hit Object or Person	11.2	12.0	18.3
2 M.V.: Rear-End	10.2	27.1	16.3
Other Configuration/Unknown	9.5	11.0	22.6
2 M.V.: Left Turn Across Traffic	8.3	10.1	8.7
2 M.V.:Sideswipe (Same Direction)	4.7	9.1	14.2
1 M.V.: Ran Off Right Shoulder	2.6	4.5	2.5
1 M.V.: Ran Off Left Shoulder	2.1	3.3	1.6
2 M.V.: Pass on Left of Traffic	1.2	1.6	2.0
2 M.V.: Pass on Right of Traffic	0.9	1.8	3.3
Total	100.0	100.0	100.0

Note: 1 M.V.: One motor vehicle, 2 M.V.: two motor vehicles.

Figure 1.2: Percentage distribution of collisions involving heavy trucks by collision configuration, 1994-1998 Average [3]

1.2.5 Driver Psychological Behavior

Truck drivers experience tremendous stress when out on the road dealing with poor drivers, heavy traffic and long hours behind the wheel. According to Stuster [12], in the United States, over speeding results in 5.9% of fatalities involved in large truck accidents, whereas failure to obey traffic devices represents 3.2%, reckless driving 2.1% and following too closely results in 2.1%. Figure 1.3 shows that the most frequent crash factors assigned to the drivers of passenger vehicles are, Run-off the road/lane, failure to yield right of way, unsafe speed, driving inattentively, and failure to obey traffic devices (e.g., traffic lights, stop signs, and other warnings).

Driver-Related Factors	Passenger Vehicles	Large Trucks
Ran Off Road/Lane	19.9	5.1
Failure to Yield Right of Way	14.4	4.6
Unsafe Speed	14.1	5.9
Driving Inattentively	8.7	2.9
Failure to Obey Traffic Devices	8.4	3.2
Erratic/Reckless Driving	4.3	2.1
Driving into Opposing Traffic	3.9	0.8
Ice, Water, Snow on Road	2.8	0.9
Following Too Closely	2.7	2.1
Vision Obscured by Weather	2.1	1.8
Manslaughter/Homicide	1.3	2.7
Vehicle in Road	1.0	1.0

FARS Data

Figure 1.3: Driver-related factors in fatal crashes between passenger vehicles and large trucks [12].

Unsafe Speed, Ran Off Road/Lane, and Failure to yield Right of Way, are the leading factors assigned to the operators of large trucks in fatal collisions with passenger vehicles. These figures show that not only commercial vehicles represent a risk on the roads but driving habits as well. Additionally, stress due to tight schedule and delivery time increases the possibility for commercial vehicle drivers to commit infractions. Monaco and Williams [13] reported that heavy vehicle accidents are inversely proportional to their firm sizes. In their study, they show that for firm sizes with less that 25 employees, accident rates are 18.81% whereas for companies with more than 5,000 employees, the percentage drops to 11.11%. This fact shows that smaller companies, usually driver owned, violate more often the driving regulations in order to increase benefits and compete with other similar firms.

1.2.6 Driver Rights

Governments are having trouble securing roads from heavy vehicles and monitoring them for many reasons. The right to privacy is the main point. Truck driving is considered as a respected profession, and like any other has unions that struggle to grant commercial drivers as much freedom as possible during their working hours. Crash Communicator [14] reports that the Canadian truck driver union proposed an increase of hours of work from 13 hours to 14 hours of driving per day which was refused by the government.

1.3 Literature Review

Road safety and transportation for commercial vehicles necessitate appropriate driving regulations due to their large mass and inertia along with the accident risks caused by driver's fatigue, drowsiness and over speeding. The published works and reported studies related to this investigation are reviewed and discussed in the following sections to formulate the scope of this dissertation.

1.3.1 Accident Risk Factors

Accidents occur for several unpredictable reasons, the most common being a human error. Over speeding is one of the main causes of accidents, because trucks having large masses increase their inertia by speeding and hence tend to become unstable. According to Stuster [12], trucks compose eight percent of all vehicles involved in fatal crashes in the United States in 1996, however, truck-involved crashes resulted in twelve percent of the total number of lives lost on the roads and highways. The disproportionate involvement of trucks in fatal crashes is a reflection of a fundamental law of physics, which is expressed by the following equation:

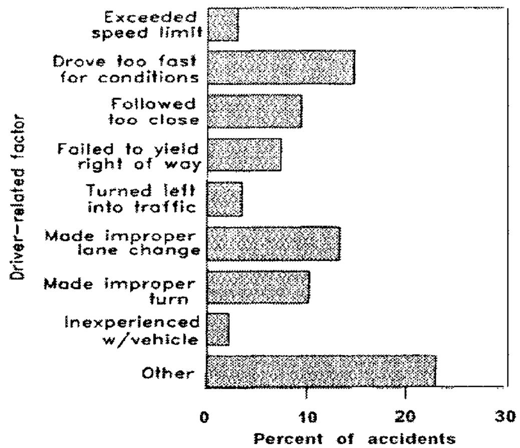
$$\text{Kinetic Energy} = 0.5 \times \text{mass} \times (\text{velocity})^2 \qquad 1.1$$

Kinetic energy is dissipated in a collision by the absorption of the energy through friction, heat, and the deformation of mass. Generally, the more kinetic energy to be dissipated in a collision, the greater the potential for injury to vehicle occupants. Trucks typically weigh 20 to 30 times as much as passenger vehicles. The structural properties and greater mass of trucks absorb better the kinetic energy generated by collisions, which poses the occupants of smaller vehicles at a considerable risk. Largely as a consequence of differential mass, the occupants of large trucks compose only 14 percent of the fatalities resulting from fatal truck crashes; 86 percent of the fatalities occur outside the truck, to pedestrians, cyclists, and, primarily, the occupants of passenger vehicles. On the other hand, overdriving with very little rest causes drivers to become fatigued and hence drowsy which substantially increases the risk of crashes that could result in death or serious injuries [1].

1.3.1.1 Speed

Transport Canada's report on heavy vehicle collisions [3] discusses rollover of trucks due to speed and cites that accidents occur when the centrifugal force produced by driving on a curve is high due to high speed. The report identifies the danger of speeding in rollover accidents of commercial vehicles, which represents 3% of all truck crashes and 13% of fatalities in the United States. It also proposes some countermeasures, among those is the RSA (Roll Stability Advisory System), which assists drivers in maintaining safe speed on curves. The European Commission report [20] cites that by reducing the average vehicle speeds by 5 km/h, the number of deaths could be cut by 1,000 a year. Gibons [16] in his report classifies speed as being prominently among driver-related factors in accidents. He adds that 45% of the time in the National Accident Sampling System (NASS) from the years 1981-85, the most frequent accident causes cited were driving too fast, poor lane changes, and following too closely (See figure 1.2). An analysis of heavy truck at-fault collision report in Oregon indicates that the principal accident causes cited are improper maneuvers, speeding, driver fatigue and inattention. Motor Carrier Safety Assistance Program (MCSAP) records show that several states have found excessive speed to be the most frequent human factor involved in accident causation. For example, Maryland, Massachusetts, Washington, and Oregon cite speeding as the most common accident causation factor. The frequencies of types of driver error for truck drivers in truck accidents in Washington State are shown in figure 1.4 [16]. Areas of poor performance by the truck driver include inattention, exceeding reasonable speed, following too closely, and improper turning maneuvers. Stuster [12] reports that it takes much longer, about two seconds, for trucks to halt compared to cars and that a car traveling at 55 miles per hour can stop in about 225 feet whereas a large truck traveling at the same speed can take more than 400 feet to stop.

Figure 1.2 —Driver-Related Factors as Cited in Heavy Truck Accident Reports



SOURCE: Office of Technology Assessment, 1988; based on National Accident Sampling System data, 1981-85.

Table 1.3 —Accident Causes Assigned in State of Washington Truck Crashes in 1984

Causal factor[a]	Number of times assigned	Percent of accidents
Driver errors:		
Inattention.	1,128	22
Failure to yield right-of-way	513	10
Exceeding reasonable speed	670	13
Alcohol	56	1
Disregard stop sign/signal	58	
Following too closely . .	277	5
Exceeding stated speed	55	1
Over center line	120	2
Improper passing	71	1
Improper turn	271	5
Apparently asleep	62	
Drugs	1	1 b
Failed to signal	22	—
Disregard warning sign/signal	25	—
Improper parking location	46	—
Improper signal	10	—
No lights/failed to dim	8	—
Deficient equipment	343	7
Other violations	606	12
No violation	1,674	33
Total accidents.	5,051	

[a] The number of causal factors does not equal the number of total accidents because several causal factors are assigned in some accidents

[b] Less than 1 percent.

SOURCE: Office of Technology Assessment, 1988, based on data from the Washington Utilities and Transportation Commission, and the National Highway Traffic Safety Administration.

Figure1.4: Several bad driving habits [16]

1.3.1.2 Driver's Fatigue

Fatigue, or tiredness, concerns the inability to continue an activity, generally because the activity has been going on for "too long". There are different kinds, such as local physical fatigue (e.g. in a skeletal or ocular muscle), general physical fatigue (following heavy manual labor) or "central nervous" fatigue (sleepiness) [17]. Fatigue slows reaction time, impairs judgment and increases the risk of collision. Driver fatigue has long been one of the major risk accident factors in commercial vehicles whose responsibility often involves irregular schedules and many driving hours. Transport Canada [18] conducted a survey on 107 related crashes and revealed that 58% were considered to have fatigue as a probable cause and 70 out of the 107 crashes reported, occurred between 22:00 and 08:00 hours, among those were 74% fatigue related accidents. In addition they reported some evidence of cumulative fatigue across days of driving due to the lack of sleep and long working hours. Transport Canada also cited in their report that night driving enhances by far driver fatigue and that fatigue is not age related. Transport Canada [3] reviewed the number of collisions, vehicles involved and casualties resulting from heavy truck collisions for different types of trucks over a five-year period ranging from 1994 to 1998. They concluded that over an average of 43,843 collisions involving heavy trucks each year, more than 80 % of the accidents were fatigue related which is also the second most frequently recorded driver condition. Haworth et al. [19] reported in their study that fatigue of car or truck drivers is a contributing factor for 9.1% to fatal accidents involving Victorian trucks in Australia. In figure 1.5, Coroner judged some factors as contributing to fatal accidents. ROSPA [20] conducted a literature review concerning fatigue in several countries and found that in general driver fatigue in commercial vehicles represents 30-40% of all accidents. The European Council [17] reported research shows that driver fatigue is a significant factor in approximately 20% of commercial road transport crashes. In France for example, a recent study showed that 10.5% of heavy vehicle drivers stated that fatigue had contributed to their road crash involvement.

Contributing factor	Percentage of accidents
No factor identified	3.2
Inattention	25.3
Fatigue	**9.1**
Speed	22.0
Alcohol	14.0
Careless driving	9.1
Wrong side of road	15.1
Failure to yield right of way	11.8
Drug driving	3.2
Suicide	4.8
Other factors	23.7

Figure 1.5: Factors judged by Coroners to have contributed to fatal accidents in Australia [19]

1.3.1.3 Driver's Drowsiness

Driver drowsiness is an important cause of truck crashes and is directly related to fatigue. Transport Canada [18] reports that a study conducted on heavy vehicle drivers revealed 54% observing drowsiness. This state was judged by two factors; the first being face video recordings from a camera installed onboard and the second being physiological measures like body temperature. The European Council [17] conducted a study to know what proportions of drivers have fallen asleep at the wheel. The results are shown in figure 1.6.

Drowsiness at the wheel	Yes
Hamelin, FRANCE, 1993 and 1999: "have you ever during your career blanked-out or dropped off for a moment?" - Long-haul lorry drivers employed by haulage firms (N = 345) in 1999 - Long-haul lorry drivers employed by haulage firms (N = 212) in 1993	 62% 58%
Van Ouwerkerk et al, HOLLAND, (N = 650) in 1986	60%
Fuller, IRELAND, (n=44) in 1978	45%
Linklater, AUSTRALIA, (n=615) in 1977	60%
Tilley, the USA, (n=1500) in 1973	64%

Figure 1.6: Percentage of driver drowsiness by European Council [17]

In a study conducted in Australia [21], the authors show that inadequate sleep is the main reason for drowsiness on the wheel and reveals that 20% of drivers admit to having fallen asleep. As well, 30% of all vehicle accidents attributed to fatigue and inattention are mainly caused by drowsiness. Hartford [22] reported from a survey conducted that drivers were detected driving while drowsy about 5% of the time. Fourteen percent of the drivers accounted for 54% of all observed drowsiness episodes. Monaco and Williams [13] showed that knowing that truck driving is a dangerous job, safety could still be improved by changing key determinants such as hours of sleep. An additional hour of sleep for truck drivers averaging 4.78 hours of sleep per day decreases the probability of violating the logbook by nearly 4%. In addition they sited that 56% of the drivers conducting the study had at least 1 six-minute interval of drowsiness while driving.

1.3.2 Implemented Solutions

Governments are striving to find ways of reducing road accidents. Although many rules and regulations are imposed on commercial vehicles, drivers always find a way to bypass the system and get away with it. Logbooks, tachographs and many other devices are implemented, but never to be as robust and unbreakable as intended. Despite their weaknesses, monitoring devices are still, in many cases, the only way of determining infractions and responsibilities in case of accidents.

1.3.2.1 Rules and Regulations

Unions, and the millions of transportation workers that they represent have long been concerned about highway safety issues. Truck driving being a dangerous job, which represents a constant risk to road safety, has pushed governments around the world to take initiatives to enhance safety and reduce accidents. Japan, Australia, Europe, the USA and Canada are the most active countries in trying to implement safer road environments. In Canada for example, the provincial ministries of transportation [18] implemented rules and regulations to limit driving speed and time for truck drivers in attempt to reduce truck accidents, which are known to be fatal in many cases. According to the Motor Vehicle

Transport Act [23], a driver can accumulate up to 13 hours of driving time following at least eight consecutive hours of off-duty time, or 15 hours of on-duty time following at least eight consecutive hours of off-duty time. In addition, drivers can be on-duty up to 60 hours during a period of seven consecutive days or after having accumulated 70 hours of on-duty time in a period of eight consecutive days. On the other hand, the Commercial Motor Vehicle technical summary compares between the U.S. and Canada [18] and reveals that U.S. rules, which are stricter than Canadian rules have a driving limitation of 10 hours, on-duty time limitation of 10 hours, on-duty time limitation of 15 hours, a minimum off-duty time of 8 hours, and 60 hours accumulated on-duty time during a 7 day period. Figure 1.7 summarizes the key aspects of U.S. and Canadian hours of service regulations.

HOURS-OF-SERVICERULES	U.S.	Canada
Driving time limitation	10 hours	13 hours
On-duty time limitation	15 hours	15 hours
Off-duty time minimum	8 hours	8 hours
7-day on-duty time limitation	60 hours	60 hours

Figure 1.7: Comparison of U.S. and Canadian hours-of service rules [18]

1.3.2.2 Monitoring Devices

The logbook is one of the first attempts to monitor the driving behaviour of truck divers. Its aim is to provide a record of a driver's work activity and enable enforcement officers to check compliance with the driving hours requirements [24]. Jackson [25] reported that logbook infractions are the most common cited area of abuse by truck drivers due to the lack of inspectors and the ease in tampering with the device. He continued on stating that the implementation of electronic on-board recorders to monitor hours of service is the best way to ensure compliance with the imposed truck driving regulations.

The tachograph, a device that graphically records the time and engine speed, was voted as mandatory on commercial vehicles [26] in Europe and TachoLink [27] is a digital tachograph that records vehicle data. Menig and Coverdill [28] reported on a digital tachometer that monitors vehicle driving to increase safe driving by measuring the driven hours, rest time, speed, RPM, heavy braking, fast acceleration, as well as an accident report. The recorded data, which has a non-volatile memory capacity ranging from 512 KB to 1 MB, can be viewed on a liquid crystal display (LCD) or by transferring the stored data to a computer. Many companies are developing such devices like Stoneridge Electronics [29] which is offering a tachograph chart recorder that is intended to record off-duty or on-duty activities during a 28 days period in addition to the speed, time and acceleration data.

According to "Autotap" [30], On-Board Diagnostic systems (OBD) are being used in most cars and light trucks on the road today. During the 70's and early 1980's, manufacturers started using electronic means to control engine functions and diagnose engine problems. This was primarily to meet the Environmental Protection Agency (EPA) emission standards. Through the years, on-board diagnostic systems have become more sophisticated. OBD-II, a new standard introduced in the mid-'90s, provides almost complete engine control and also monitors parts of the chassis, body and accessory devices, as well as the diagnostic control network of the car.

Menig and Coverdill [28] proposed a device that operates similar to a black box, and which can record several truck parameters such as speed, driving hours, braking and acceleration. Dole [31] discussed the use and implementation of the new on-board recorders in heavy vehicles, stating that with today's technology, everything that drivers do in their job can be recorded in real-time. The latest versions of such devices can provide drivers immediate feedback when they are exceeding pre-set safety parameters. Grace et al. [32] conducted a study on the prediction of drowsiness on truck drivers through the use of an onboard camera that measures eye closure.

1.3.2.3 Control Systems

Ukawa et al. [33] conducted a study aiming at realizing autonomous driving along a given test course by using a linearized model of longitudinal and lateral actuators and controllers to control speed and lane tracking. Their experiment aimed at reducing and preventing traffic accidents by using a heavy-duty truck equipped with traffic congestion warning and a controller. Haworth et al. [19] proposed an eye-closure monitoring device that continually compares the relative reflectiveness to light of the eyelid and eyeball with the help of an electronic processor. An alarm is activated approximately 0.5 seconds after the eyelid begins to shut should the eyelid not open in time since psycho-physiological measures the eyelid motion to be around 0.5 seconds.

1.3.3 Smart Cards

The Smart Card, a term coined by French publicist Roy Bright in 1980, was invented in 1967 and 1968 by two German engineers, Jürgen Dethloff and Helmut Gröttrupp who filed for a German patent on their invention in February 1969 and were finally granted titled "Identifikaden/Identifikationss-chalter", in 1982. Independently, Kunitaka Arimura of the Arimura Technology Institute in Japan filed for a smart card patent in Japan in March 1970. In the following year, in May 1971, Paul Castrucci of IBM filed an American patent titled simply "Information Card", and on November 7, 1972, was issued U.S. Patent 3,702,464. Between 1974 and 1979 Roland Moréno, a French journalist, filed 47 smart card-related applications in 11 countries and founded the French company Innovatron to license this legal tour de force [34].

Nowadays, the Smart Card, is a credit card sized plastic card embedded with an integrated circuit chip. It provides not only memory capacity, but computational capability as well. The self-containment of smart cards makes it resistant to attack, as it does not need to depend upon potentially vulnerable external resources. Because of this characteristic, smart cards are being used in several applications that require strong protection, and authentication as stated by Chan [8]. Different applications require different types and features of Smart Cards.

1.3.3.1 Card Types

Many types of cards have been implemented in several applications ranging from the simple telephone card to the highly secured credit cards. Magnetic-strip cards are cards similar to the famous debit card used in financial applications. It is read by pulling it across a read head, either manually or automatically, whereby the data is read and stored electronically. Drawbacks of this type of card are its low memory storage capacity of only 1000 bits and the ease in altering the stored data.

The Smart Card is the youngest and cleverest member of the family of identification cards. Its characteristic feature is an integrated circuit embedded in the card, which has components for the transmission, storage and processing of data. Data transfer can take place either via the contacts on the surface of the card or via electromagnetic fields, without any contact use. This type offers advantages compared to the previous one, among those is a much larger memory storage capacity and most importantly, high protection of the stored data against unauthorized access and tampering. This together with the ability to compute cryptographic algorithms allows smart cards to be used in the implementation of convenient security modules that can be carried about at all times. Smart cards are divided into two groups, according to their differences in both functionality and price: memory cards and microprocessor cards [35]. The first cited is usually an EEPROM memory card controlled by the security logic. There exists also memory chips with more complex security logic, like encryption. The microprocessor card is, as the name implies, a processor that is surrounded by a ROM, EEPROM, RAM and an I/O port. This type of card contains a much greater memory than the previous one stated and operates very much like a small computer [34]. Contactless smart cards do not need any physical contact for them to be accessed. The communication is done by ways of electromagnetic radiation, essentially working at Radio Frequency (RF) frequencies. Advantages for this type of card is that it operates at a variety of distances between it and the reader. Optical memory cards are used where the storage capacity of smart cards is insufficient because they can store several megabytes of data. However, with the currently available technology, these cards can be written to only once. Chaum [36] stated that smart card prices range roughly between US$ 0.5-2.00. This estimate was determined based on the unit card production price as well as its duration and frequency of use. Memory card chips are much smaller and consequently much less expensive to produce than microcontroller cards. Hence, their prices roughly range between $US 0.1-0.4 depending on the purchased quantity.

1.3.3.2 Application Fields

Smart card chips have found many useful applications in the market around the world like public card phones in Germany, which were implemented in 1989 by the German state telephone company [35], Telekom. By the end of 1994, smart cards operating as health insurance cards were issued in Germany and in some countries it has become nowadays very common for smart cards to be used as toll payment on roads. According to Rankl and Effing [35], the GSM (Global System of Mobile Communication) network, originally a European standard for mobile telephony has become a world wide recognized system operating in more than 120 countries with more than 120 million subscribers using smart card chips in cellular phones. Hall et al. [37] implemented the 'Poket Doctor', which is a device that makes use of the smart card to store patients' personal medical information. It is carried by individuals in their wallets and can be accessed by medical personnel directly on the site of an accident. Nakamura et al...[38] made use of smart cards for toll collection in Japan. Itoi and Honeyman [39] developed a practical security system based on Smart Cards. Rankl and Effing [35] consider the payment systems today as the primary market sector, namely electronic payment systems because smart cards are by nature particularly suitable for financial transaction applications due to their ease and secure data storage. Lately, smart card applications have been found in the trucking industry. In its press release, Giesecke & Devrient Company stated that as of July 2004, European truck drivers will have a smart card to authenticate themselves to the new digital tachographs in place before starting their journey. The smart card will incorporate a photograph and the driver's personal details and will be kept inside the tachograph's card reader throughout the trip.

1.4 Scope of the Present Work

From the literature review, it is apparent that heavy vehicles represent a constant accident risk due to their high dynamic instability as well as the human error factor. Among the safety issues, fatigue, drowsiness and speed have been the major contributors of road accidents. Long driving time followed by inadequate rest often leads to risk related conditions. This has led to the development of onboard data recorders to reduce crashes by monitoring drivers during their journeys. Examples of such device, the Electronic Control Module (ECM), the Black Box and the tachograph have been the most popular in the automotive industry up to recently, where many fraud and tampering attempts have been reported and the need for better data recorders has risen. The present work attempts to address the above issues.

1.4.1 Objective of the Present Work

The overall objective of the thesis research is to contribute to the enhancement of road safety through monitoring truck drivers and their vehicles. Achieving this task is done by designing and developing a new data recorder capable of recording safety related parameters onboard heavy vehicles. The specific objectives of the thesis are:

1. To develop and implement an inexpensive simple data recorder capable of monitoring speed and driving time in a truck as well as competing with the present ones.
2. To implement smart cards as a data storage for the recorded driving information and as an identification card for truck drivers.
3. To integrate various communication protocols to the system in order to connect it with external sensors and computers making it versatile.
4. To investigate the capability of the system to record other information such as the ones related to the driver's driving habits or dangerous manoeuvres.
5. To investigate the capability of the system to record conditions of the vehicle that may lead to future accidents.

The work related to this thesis is in no way related to establishing limits and thresholds for the safety of the vehicle or driver, but to evaluate the capability of the proposed system to measure or detect these limits.

1.4.2 Organization of the Thesis

In chapter 2, four bad driving habits are identified as contributors to road accidents, namely drowsiness, safe spacing between vehicles on the road, effect of speed on road safety, and the lane changing maneuver. The effect of falling asleep on the wheel during a trip is discussed trying to identify means of detecting and preventing it. Safe distance spacing and speed effect on road safety are equally discussed and an attempt to find a safe lane change trajectory domain is undergone for different speeds. In addition, data recorders are briefly investigated showing their advantages and disadvantages as well as the several frauds and tampering attempts to which they are subjected.

The proposed data recorder is introduced in chapter 3, discussing the major parts that constitute the system and the settings used to simulate its operation.

In chapter 4, the smart card is presented in details. The physical and electrical properties are explained as well as its operating system. The card personalization is discussed and developed to suit the application. The memory organization of the chip is also presented and the file structure is designed to maximize data storage. The communication protocols used in the smart card technology are investigated and explained as well as the several commands and responses that are needed in the context of this dissertation.

In chapter 5, the data processor is designed and developed. An electronic circuit layout is presented to show the several electronic components and connections used for the design. The microcontroller implemented in the system is explained in detail as well as the code generated for the application. The interface with the external sensor and the computer used for speed-data simulation is developed using an analogue to digital converter and the parallel communication protocol. A serial communication is also integrated as an interface between the data processor and the smart card. The results generated by simulation, are shown in the last section of the chapter.

The highlights of the study and the conclusions drawn are summarized in chapter 6. The recommendations for future development of the monitoring system are presented.

CHAPTER 2

ONBOARD MONITORING SYSTEMS AND THE IMPORTANCE OF THE RECORDED DATA

2.1 Introduction

Much progress has been made over recent years in understanding many aspects of driver-vehicle interactions, for example, linked to the influence of improved safety systems and interaction with Intelligent Transportation Systems (ITS). Airplanes have been fitted, for years, with flight-data recorders that would help crash investigators determine what happened in the seconds before a crash. The automobile industry has recently realised the benefits of such systems for vehicle safety and many monitoring devices such as the Black Box or the tachograph proved to contain invaluable information. Although electronic sensors gained wide use in production automobiles in the 1970s, it is not until recently that crash-recorders were implemented. Nowadays, vehicles have reached a very high degree of sophistication and with each new electronic system installed, new sensors ranging from the wheel speed sensor to the vehicle yaw rate sensor are required. This has helped monitoring devices reduce their installation and operation cost due to the need for fewer new sensors.

The importance of the recorded data is valuable in determining the causes of an accident. Speed sensors as well as yaw pitch and roll rate sensors can play a role in preventing road accidents. The measurements are compared with certain set threshold values to indicate a dangerous maneuver or a vehicle state that could lead to an accident and inform the driver about it. Commercial vehicles are first targeted due to their long hours of operation, their high mass and relative instability on the road, which makes them a potential risk for accidents.

This chapter discusses the relevance and importance of the measured data for road safety and prevention of accidents emphasizing on the impact of bad habits. It also investigates some monitoring devices onboard trucks such as the tachograph and the Black Box and introduces the smart card recorders.

2.2 What would be a Bad Driving Habit?

Heavy vehicle driving is a dangerous business where many factors play a determining role during a journey and affect it in a positive or negative way. Economical pressures exerted by a client or an employer, road conditions, weather conditions, etc. affect the driver and somewhat forge in him bad habits that can present a risk to road safety. Two questions arise; what is a bad driving habit? And, how can it be detected?

This dissertation does not attempt to solve this complex issue but rather tries to clarify the meaning of a bad habit and proposes measures of detection that could probably help prevent it in dangerous circumstances. The system designed in this thesis is expected to detect some risky maneuvers that are often encountered with heavy vehicle drivers.

2.2.1 Drowsiness

Falling asleep on the wheel is one of the major causes of road accidents involving heavy vehicles. This usually occurs during the early hours of the morning or during afternoons and is caused by fatigue due mainly to excessive driving and short periods of rest that are, several times, ignored due to a late schedule or proximity of a destination.

This human error, which reveals to be fatal on the road in many instances, is the result of carelessness by the driver and a bad habit of ignoring rest periods. Although data recorders are being installed in trucks to monitor drivers and thus try to reduce road accidents caused by drowsiness, truck drivers are still finding ways to outsmart the implemented systems. The device developed as part of the thesis, aims at preventing drivers from falling asleep, regardless of the system's correct operation. A study conducted by King and Mumford [15] linked drowsiness detection to the steering wheel movements using a special algorithm.

An experiment was developed and conducted at Concave Research Centre where a yaw rate sensor is installed on the shaft of the steering wheel thus monitoring its motion. During normal driving conditions, a driver always tends to move the wheel, even while driving on straight roads. Figures 2.1-a to 2.1-d and 2.2-a to 2.2-d illustrates the result of an experiment recorded on a portion of highway using a passenger car. A subject drove a car on a trajectory in the city for a period of 30 minutes and on the highway for 6 minutes. A portion of 90 seconds of the highway driving is selected for analysis because it represents the driving pattern required for the study, which is the most suitable since it represents a portion of open road. Figure 2.1-a shows a small motion detected by the gyroscope along the straight path. The habit of constantly moving the steering wheel can also be associated to truck drivers due primarily to being a human factor condition regardless of the vehicle driven. A spectrum analysis is performed to the signal in order to detect a drowsy condition characterized by no motion of the steering wheel and a constant output from the gyroscope describing noise. The data is measured with a sampling rate of 256 samples per second, saved in a data logger and analyzed using the

DADisp software. The analogue sensor output is stored in a digital form using a 16-bit A/D converter with a 10V DC operating voltage. In order to restore the initial gyroscope signal for analysis, the digitized signal is hence, divided by 2^{15} and multiplied by 10 to account for the modifications imposed by the analogue to digital converter as shown in figure 2.1-c. A spectrum analysis shown in figure 2.1-d is done to the resulting signal in order to identify the dominating frequency. The result, not being adequate for the application, a moving average of the spectrum analysis is done to clarify the analyzed signal and determine meaningful frequency peaks. The results obtained with the moving average analysis using '7','8' 9' and 10 elements revealed several peaks in the range of 0.25 to 1.5 Hz with a maximum peak at 0.5 Hz as shown in figures 2.2-a through 2.2-d respectively. This could suggest that the driver is slightly adjusting the steering wheel from 4 motions per second to 3 movements every 2 seconds. This can help prevent any drowsiness during driving by implementing the results on an onboard recorder that monitors the drivers and triggers an alarm when out of range frequencies are detected over a predefined period of time

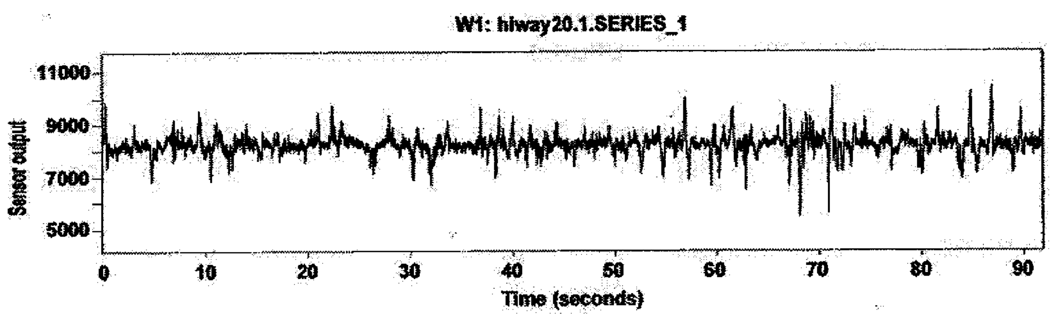


Figure 2.1-a: Gyroscope signal in data logger

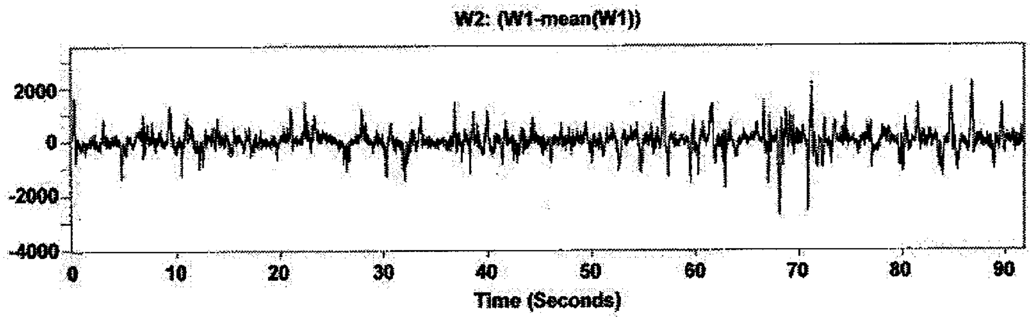


Figure 2.1-b: Signal in data logger without its mean value

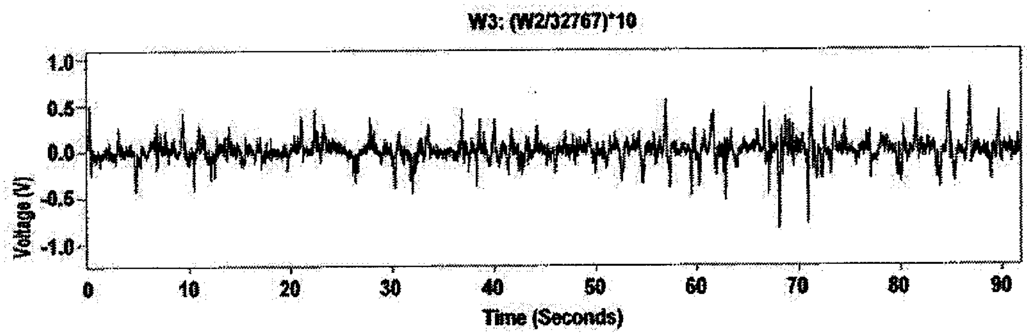


Figure 2.1-c: Gyroscope signal without the effects of the A/D converter

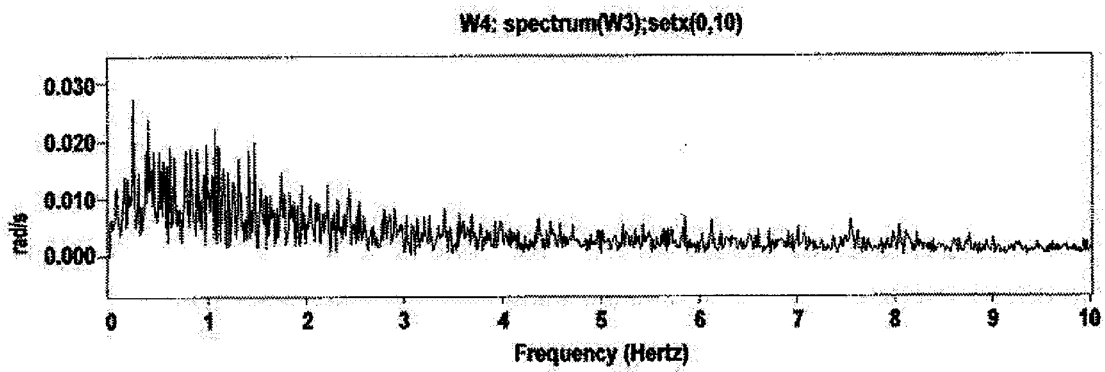


Figure 2.1-d: Spectrum Analysis of the cleaned signal

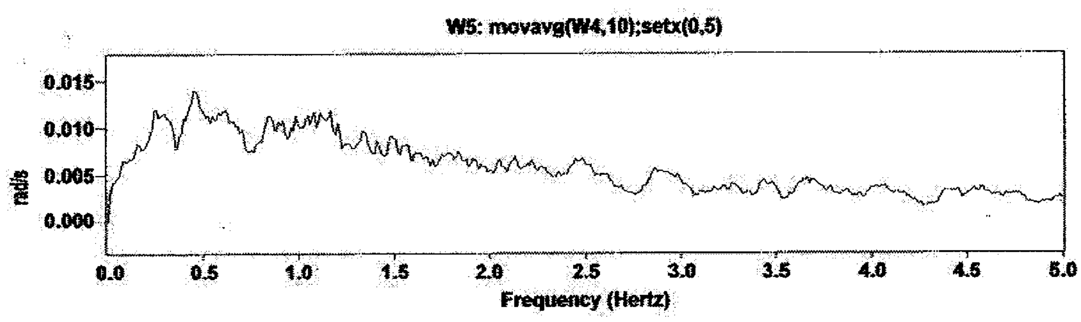


Figure 2.2-a: Moving average with 10 elements

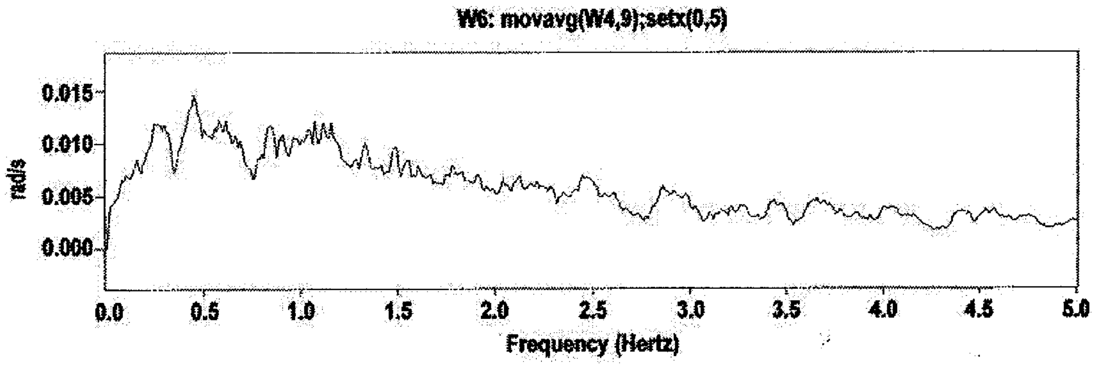


Figure 2.2-b: Moving average with 9 elements

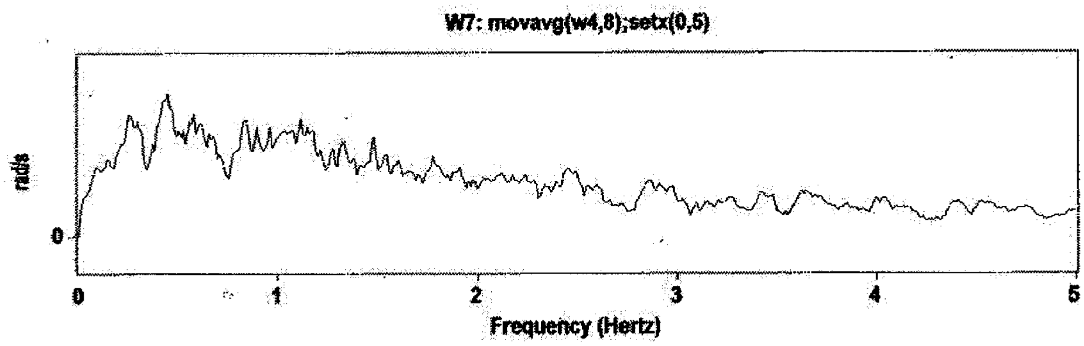


Figure 2.2-c: Moving average with 8 elements

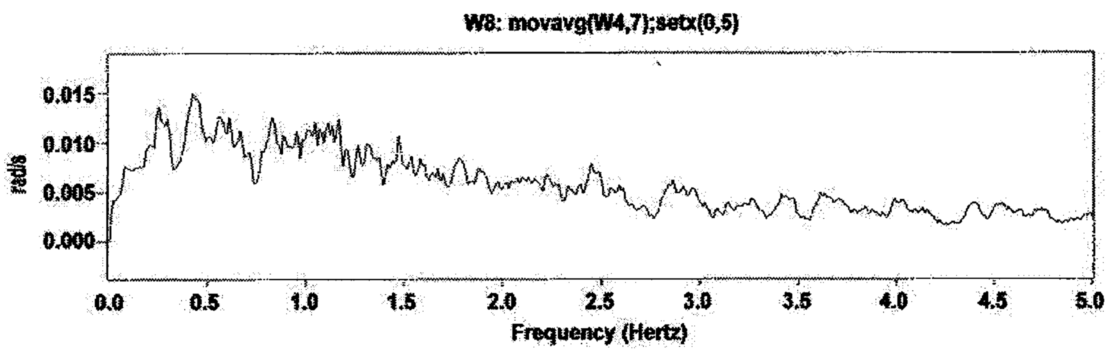


Figure 2.2-d: Moving average with 7 elements

2.2.2 Safe Distance Spacing

Keeping a safe distance with the leading vehicle is an important part of driving. The space varies with different types of vehicles, driving speed, road condition, etc. Some drivers have the bad habit of following a vehicle too closely, which represents a potential accident risk. Since heavier vehicles require longer stopping distances, they need a greater safe spacing between them and the leading vehicle. Using TruckSim, the stopping performance of a 3-axle/2-axle tractor semi-trailer was simulated. The truck moving at constant set velocities brakes with 0.7 MPa of decelerating pressure, the results are shown in figure 2.3. The simulation is conducted on a straight road and the braking distance is measured from the time the brake is applied until the truck is stopped.

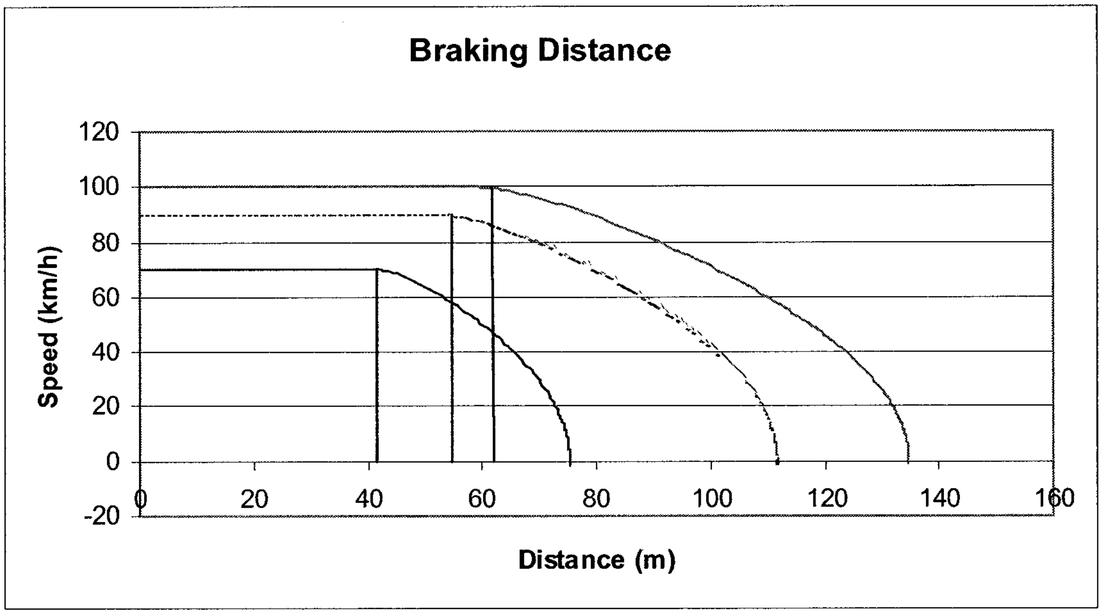


Figure 2.3: Braking distance of a 3axle/2axle tractor semi-trailer.

The truck moving at 100 km/h needs a stopping distance of around 70 meters whereas driving at 70 km/h requires only around 35 meters. This clearly shows that reducing the speed by 30% reduces the stopping distance by 50%, which is in many cases could mean the difference between a crash and a clean stop.

Assume as an example that the 3-axles/2-axle tractor semi-trailer is following a car on a straight dry road with the two vehicles traveling at a constant speed of 90 km/h. If the leading car brakes suddenly the truck is expected to brake after a certain time and the truck velocity profile in figure 2.3 shows the curve for the decreasing velocity. From the graph it can be said that for a speed of 25 m/s (i.e.90 km/h) and according to simple kinematics the truck deceleration can be calculated as:

$$A = \frac{V_f^2 - V_i^2}{2 * \Delta x} = \frac{0 - (25)^2}{2 * (55 - 110)} = 5.68 m/s^2 \qquad 2.1$$

Calculating the safe distance (SD) according to Huang [40]:

$$SD = \frac{V_2^2}{2A_2} - \frac{V_1^2}{2A_1} + V_2 * t \qquad 2.2$$

where V_1, A_1 and V_2, A_2 are the velocity and deceleration of the leading car and the following truck, respectively. The variable t is the reaction time of the following driver, which is assumed to be 1sec when both vehicles are driving on the highway at 90km/h. The car deceleration is assumed to be 6.5 m/s^2 for dry road surface according to Huang [40], and the safe distance is calculated to be SD = 32m. In comparison to two vehicles following each other by Huang [40], SD = 25m which is 1.5 times less than the stopping distance of the truck. It is therefore very important to keep track of such a bad habit in truck driving.

2.2.3 Effects of Speed on Road Safety

Speed, as described in the literature review, is one of the leading causes of road accidents. The effect of speed on cornering stability, braking distance and impact forces increases at the square of the speed increase. Brooks [41] indicates that 23% of heavy vehicles exceed the posted speed limit and 3.2% of trucks exceed the limit by more than 10%. The over speeding habit is common among drivers and has a direct effect on road safety. Speed increases the risk of accidents and speeding trucks become more vulnerable to evasive maneuvers like jack-knifing. This situation is more common with semi-trailer combinations but does also happen to full trailers. In a jack-knife, the drive axles locks-up under braking before the steer axle locks up and any small lateral force imbalance due to road slope or surface condition will cause the drive axles to move rapidly out of line [42]. The main cause for such a situation may be due to a poor load balance, harsh braking or braking on corners, particularly on wet or icy roads.

Another effect of speed in road accidents is in cornering. A heavy vehicle trying to undertake a turn at a high speed has a good chance of rolling over, mainly due to its elevated center of gravity. With reference to figure 2.4, the equilibrium equation for roll moment about a point on the ground at the center of the track is given by the equation:

$$M * h * a = \frac{(F_2 - F_1) * T}{2} - W * \Delta y \qquad 2.3$$

where M = truck mass

W = truck weight

h = height of the center of gravity

a = lateral acceleration

F_2, F_1 = load reactions on the wheels

T = distance between the center of the vehicle and one of the tires

Äy = displacement travelled by the center of gravity

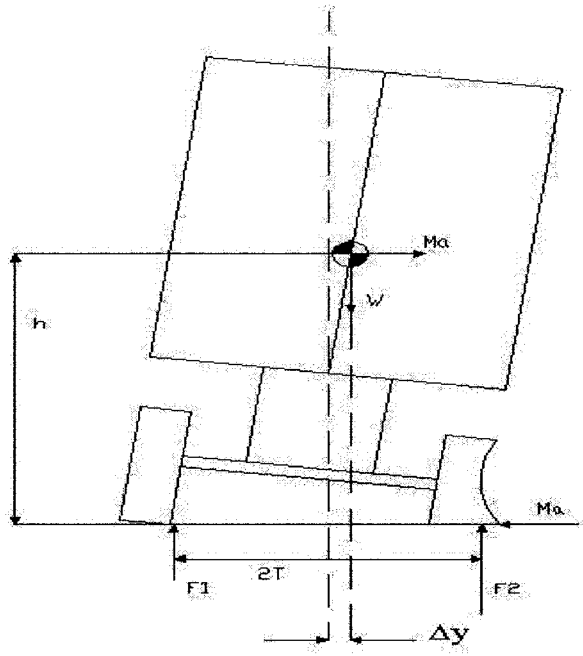


Figure 2.4: Rigid vehicle roll model

A simulation using TruckSim shows the lateral accelerations of a 3-axle/2-axle tractor-semitrailer traveling at 3 set speeds during a cornering maneuver. Figure 2.5 illustrates the lateral acceleration of the tractor at its C.G. The black dotted curve represents the 100 km/h speed test, where the simulation stops after the 5 seconds mark due to rollover of the truck. Figure 2.5 also shows the trailer lateral acceleration, which rolls over at the same speed. The graphs show that at a 100km/h, the trailer lateral acceleration peak of 0.38g provoked a rollover of the vehicle whereas lateral accelerations below the predefined value of 0.33g [71] did not lead to a crash. TruckSim assumes vehicles with suspension which characteristics are similar to those of the real vehicles.

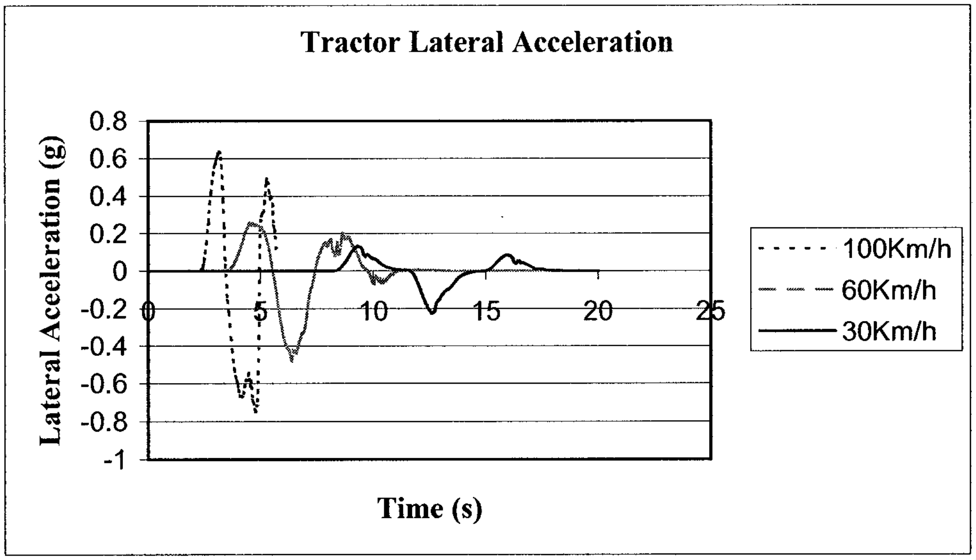


Figure 2.5: Tractor lateral accelerations

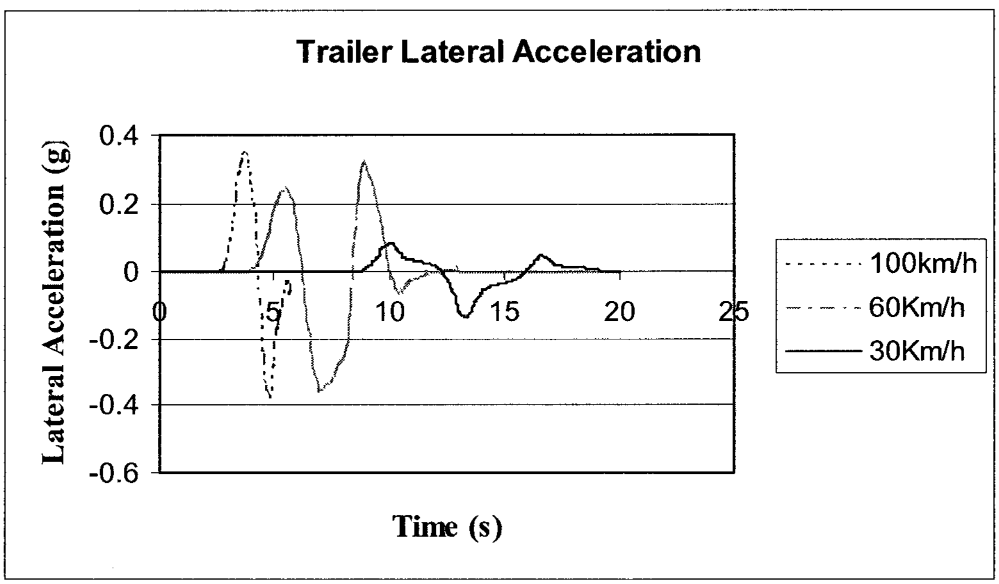


Figure 2.6: Trailer lateral accelerations

2.2.4 Lane Change

Lane change is a complex maneuver involving several parameters. A vehicle, changing lanes, needs to verify the proximity of the leading and following vehicles if any, evaluate the distance and time needed to proceed and go along a safe and correct path thus limiting any swerving or unwanted critical lateral accelerations that could provoke rollover. Speed and mass being the essential variables in such a maneuver, it is unsurprising that extra attention is focused on heavy vehicles rather than on automobiles.

An investigation of the lane change trajectory is done in order to delimitate a certain domain in which, a truck, moving at a certain speed, would safely undergo a lane change. Using the TableCurve software, points are selected according to an assumed trajectory that describes the maneuver. Knowing that the limiting lateral acceleration for heavy vehicles is in the range of 0.32g to 0.36g, the value of 0.3g is assumed to be critical in this study. Lateral acceleration is defined as:

$$A_{lat} = \frac{V^2}{R} \tag{2.4}$$

where V is the traveling velocity and R the radius of curvature of the trajectory. The critical point in a lane change is defined according to the radius of curvature at a particular point, which is defined as follows:

$$R = \frac{[1 + (\frac{dy}{dx})^2]^{\frac{3}{2}}}{\frac{d^2 y}{dx^2}} \tag{2.5}$$

Several points describing the lane change path are chosen randomly. For a specific speed, a certain critical radius of curvature is determined according to the set critical lateral acceleration and a multitude of curves are fitted to the chosen points in order to find the optimal trajectory.

The curve-fitting optimization simulation results are shown in figure 2.7 and 2.8 for two speed values, respectively 100 km/h and 70 km/h. Each speed domain is set according to the trajectory defined by equations 2.6 and 2.7.

$$y = a + b * \sin(\frac{2 * \Pi * x}{d} + c) \qquad 2.6$$

where a=1.72

b=1.75

c=0.714 to 4.49

d=133.09

$$y = a + \frac{b}{[1 + e^{(-\frac{(x-c)}{d}}]} \qquad 2.7$$

where a=0.14

b=3.58

c=115.77 to 25.77

d=6.56

A set of paths describing a lane change occurs over a distance of 150m on a road width of 3.5m. The equations representing 70 km/h and 100 km/h driving speeds respectively yield radiuses of curvature calculated at the minimum and maximum points of the paths. From equation 2.4, the critical radius $R_{critical}$ is derived for the 100 km/h and 70 km/h speeds respectively. Knowing that the critical lateral acceleration is 0.33g, the value of 0.3g is chosen in this study. At 100 km/h, $R_{critical}$ = 257m, whereas at 70 km/h, $R_{critical}$ is 126m.

Equation 2.6 gives R_3 = 257.4m and R_4 = 259.46m for c ranging from 0.714 to 4.49 where the subscripts '3' and '4' represent the delimiting curves indices of the 100 km/h domain.

Equation 2.7 gives R_1 126m and R_2 = 128.48m for c ranging from 115.77 to 25.77 where the subscripts '1' and '2' represent the delimiting curves indices of the 70 km/h domain.

The results found shown that the established lane change equations ensure a safe maneuver as long as the radius of curvature is greater than radius set for each of the limiting curves. The lane change trajectory of Liu [71] as well as the default path proposed by TruckSIm fit the curve domain.

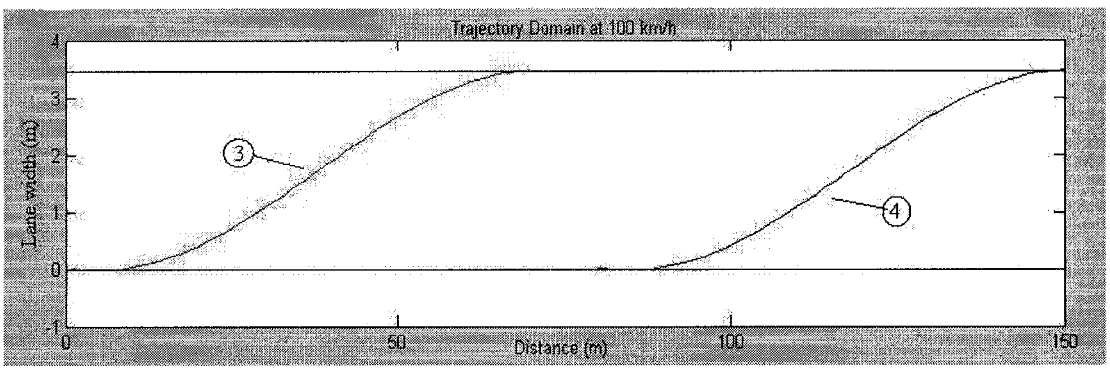


Figure 2.7: Lane Change domain at 100 km/h

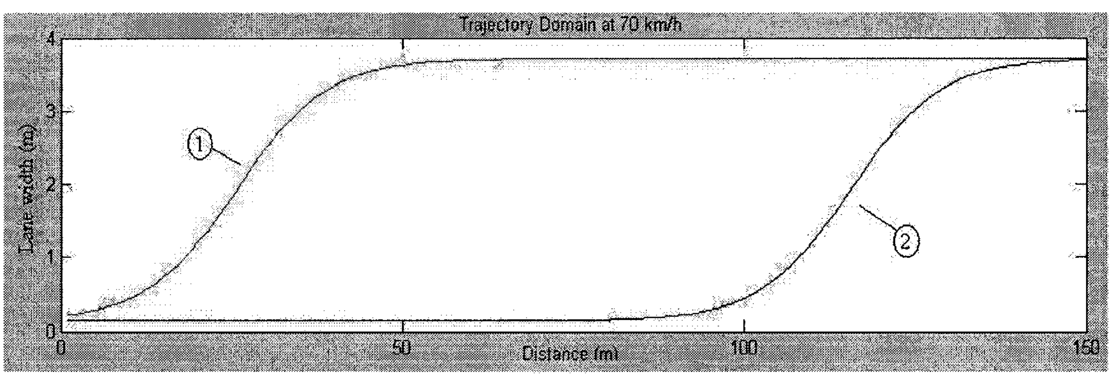


Figure 2.8: Lane Change domain at 70 km/h

The method of determining a safe lane change for commercial vehicles through the study of the path undergone is not very useful and accurate. The great number of paths that can be taken, the different speed conditions and the different equations, which describe similar tracks, make it hard to implement on a controller. A better approach to this study would be to monitor the lateral acceleration and speed of the heavy vehicle.

2.3 Onboard Monitoring and Relevance of Recorded Data

The cases of bad driving presented in the previous sections account for a large portion of road infractions leading to accidents, but there still remain unaccounted violations that will not be discussed in the context of this dissertation. The importance of the records is measured by their ability to contribute to the reconstruction of accidents, which can be crucial in many cases. An example of such relevance is when an accident happens and no eyewitnesses are present at the moment of the crash. The recorded information will help, in such a case, to determine the cause of such an event.

Data recorders are making their way into the automotive industry and proving that they play a very important role in road safety, becoming an essential part of every vehicle, especially for trucks. The onboard recorders are very diverse and range from the Electronic Monitoring Unit to the well-known tachograph. This investigation is only based on the Black Box and the tachograph due to their similarity with the monitoring system designed in this thesis. Both devices store driving information and only differ by the way the data is stored. The tachograph, widely used in the commercial fleet nowadays, stores its information on removable paper disks, whereas the Black Box uses an electronic memory and records are kept in the vehicle at all times. A third monitoring device that has not been introduced to commercial vehicles yet is the smart card recorder, which will be discussed in detail in later sections of the dissertation. It is a merger of the two traditional recorders because it stores memory on an electronic chip and permits viewing the stored data on an external computer. The devices in question are discussed in sections 2.3.1 through 2.3.3 in detail showing their advantages and disadvantages as well as introducing the smart card monitoring system.

2.3.1 The Black Box

The use of on-board electronic data recorders in the aviation industry is well known and in particular the Black Box. In the event of a crash, the recovery of in-flight recording systems is a priority for collision investigators and the data obtained has proven to be crucial for helping determine the causes of the accident and the crash reconstruction process. Event data recorders have made their way into the automotive industry a few years ago, but it's only recently that there has been a proliferation of such technology in the vehicle fleet. This is primarily due to the introduction of supplementary air bags and, in particular, due to the need to monitor and control the deployment of these systems. In addition, the Engine Control Module (ECM) implemented in vehicles in the early 1970s to monitor the engine operation has evolved with the new sophisticated electronic systems, which found their way into the automotive industry. The updated versions comprise extra sensors like accelerometers, yaw rate sensors, wheel speed sensors and several others. This has facilitated the installation of on-board data recorders and in particular the Black Box. General Motors [43-44] has started equipping some of its vehicles with pre-crash data recorders since 1999. The Sensing and Diagnostic Module (SDM) predicts the deployment decisions of air bags within 15-50 msec after impact and helps also determine when to record pre-crash data. Figure 2.9 shows how the pre-impact sensor data might appear when downloaded to a computer. The Detroit Diesel Corporation [45] introduced the Detroit Diesel Electronic Control (DDEC) in 1985 in order to increase fuel economy in trucking fleets. In 1998, the data recording capabilities including engine speed, vehicle speed brake and clutch positioning, etc. were added to the DDEC.

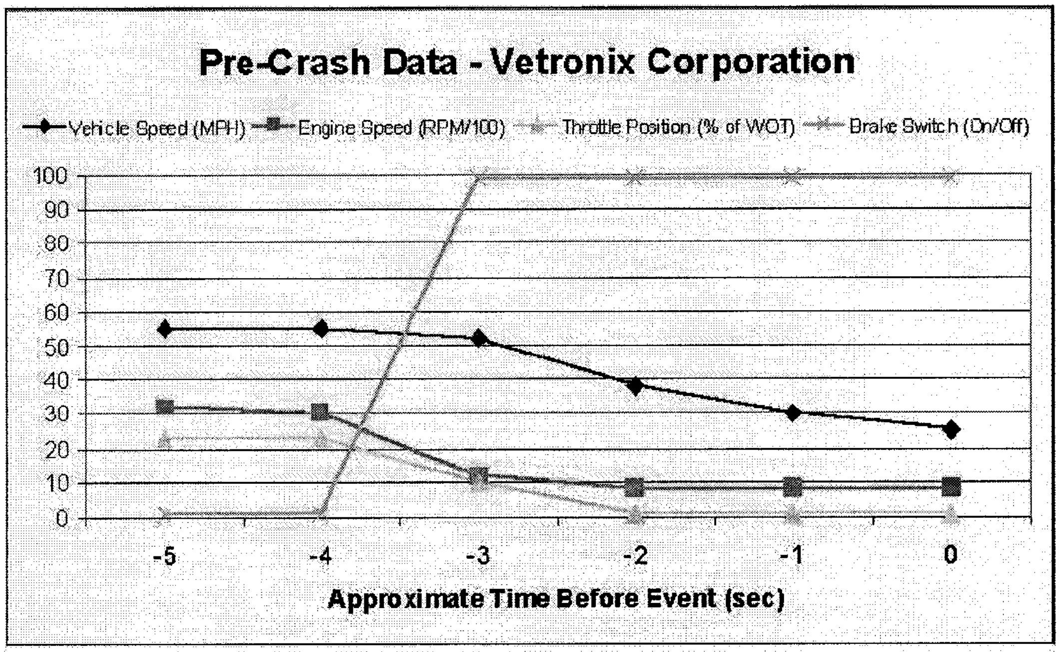


Figure 2: Pre-Crash Data vs. Time

Figure 2.9: Example of Pre-Crash Data Recorder data [43]

Siemens VDO Automotive [46] designed the USD-Black Box, a sophisticated electronic device designed to collect data only when an accident occurs. Packaged in a very small size, it can be fixed to the chassis, under the seat or in the boot of the vehicle. If an abnormal impact occurs during driving, one of its sensors triggers the registration process, which ranges from 30 seconds before to 15 seconds after the crash. The vehicle speed can be deducted from the acceleration data and compared to the wheels status thus concluding about blocked wheels and skidding.

2.3.2 The Tachograph

A tachograph is a recording device used in most heavy vehicles throughout Europe and North America to monitor the speed of the vehicle, the distance driven, the driving time, and is also used for secondary purposes ranging from accident investigation to the detection of fraud. The conventional tachograph consists of a paper disk that contains the driver's profile as well as the driving pattern. Figure 2.10 and figure 2.11 show the tachograph and the chart that is used for recording the driving information.

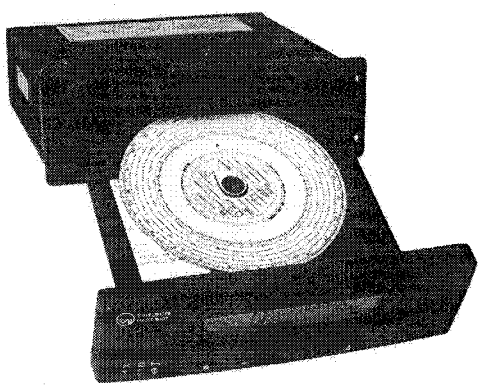


Figure 2.10: The tachograph.

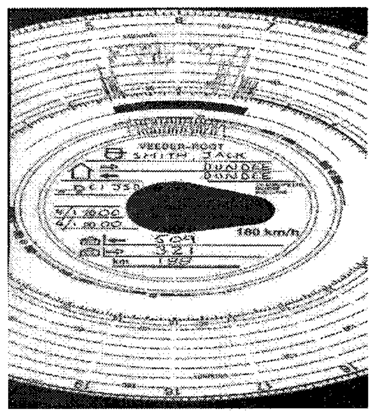


Figure 2.11: Recorded data on paper disk

(Pictures courtesy of Canadian Automotive Instruments LTD.)

The tachograph can be associated with the previously discussed Black Box because it performs similar tasks. Nevertheless, the tachograph data is recorded mechanically versus electronic recording in the Black Box and the data storage is retrieved frequently by the user which has facilitated tachograph tampering and fraud. Anderson [47] reported the main techniques used at present for tachograph offences. Major offences exploit procedural weaknesses as well as the tachograph and its supply of power. It involves:

- 'Ghosting', which consists of manipulating tachograph charts so that there appears to be more drivers than there actually are.
- Altering the clock to simulate various rest periods, inserting a fresh tachograph chart and hiding the old one, forging a chart by hand or driving without a chart.
- Forgetting to write the date on the chart center field, presenting a hitchhiker as a co-driver, using a chart that does not correspond to the driving area and filling fictitious details in the center of the chart.
- Including switches into the sensor input cable, adding an extra sensor from which measurements are taken instead of the one mounted on the vehicle's gearbox, inserting a ground wire into the cable thus shorting out the impulses and replacing the fuse with a blown one.
- Tampering with the tachograph calibration, reducing the supply voltage causes the device to freeze and blocking the speedometer by pressing on its center to prevent the needle from moving during acceleration.

It is for these several reasons that the European Union [48] has voted the implementation of digital tachographs onboard trucks. This new version comprises a smart card, which replaces the usual paper chart and holds the driver records for a period of 28 days.

2.3.3 The Smart Card Recorder

Smart cards are finding their way into the automotive world and in particular, in road safety related fields. The digital tachograph, which makes use of the smart card, is being implemented in Europe in June 2004 as announced by Giesecke & Devrient Company in a press release [48]. The driver card resembles the European driving license and is produced according to ID card standards. It incorporates a photograph and the driver's personal profile that are recorded on the chip. The intent of this card is to authenticate the drivers to the tachograph prior to the journey. A sensor in the engine compartment transmits the registered data to the tachograph in the cockpit.

In the next chapter, the solution proposed by the author at Concave Research Center [51] at Concordia University is presented. As part of the present work, a monitoring system for commercial vehicles using the smart card was developed. The details will be elaborated in the next chapters of the dissertation. This device, aiming at enhancing road safety, operates as a combination of the Black Box and the digital tachograph with extra features such as data portability, digital encoding as well as being conceived as a low cost device. Vehicle speed, steering wheel yaw rate, lateral acceleration, driving time, drowsiness status and dangerous handling maneuvers are among the parameters that are or could be monitored by the system. The recorded information is processed using a microprocessor and sent in a condensed format to a smart card with an 8K EEPROM memory. The driver's personal details are also stored on the card. The system could be additionally developed with advances in smart card technology to comprise along with the monitored driving information, a driver's license, which has already been implemented onto a card and a brief medical record of the cardholder. The smart card records are only accessed in case of inspection by enforcing authorities or after a crash. The designed system monitors lateral acceleration, speed, driving time as well as off-duty time. Nevertheless, it can also be accommodated to measure longitudinal acceleration that could be an important parameter in the reconstruction of accidents or in determining the habits of drivers.

2.4 Summary

Bad driving habits could be investigated and their impact on road safety is emphasized in order to justify the use of data recorders, which monitor truck drivers during their journeys, and show the importance of the information in the reconstruction of accidents. Suggestions about lane change maneuvers and safe distance spacing between two vehicles on the road are proposed, a method of determining drowsy behavior is investigated using a gyroscope placed on the steering column of the vehicle and the effect of speed on stability and other maneuvers is discussed. A further investigation of the onboard recorders shows the weaknesses of the current implemented technology, mainly the Black Box and the tachograph. The smart card data recorder is introduced as part of the solution to enhance road safety and limit fraud and tampering with the monitoring equipment.

CHAPTER 3

MONITORING SYSTEM OVERVIEW

3.1 Introduction

Heavy vehicles due to their excessive weights and large dimensions represent a bigger threat to road safety than any other type of road-vehicle. According to a study conducted by Transport Canada [3], heavy vehicles represent an average of 46,239 collisions per year, which is equivalent to 4% of all vehicles involved in all collisions in Canada. Although this number represents a small portion of the total number of road-vehicle accidents, it accounts for a great number of fatalities on the road. That number is decreasing every year, but is still considered high and thus there is need for safety-related devices onboard heavy vehicles.

The solution presented in the dissertation consists of a monitoring device, which is a combination of the tachograph and the Black Box that are installed onboard heavy vehicles. The system's task is to monitor and record the driving time, speed and rest time of the driver, as well as the lateral acceleration or the number of steering wheel movements per minute, which shows the state of drowsiness of the driver. The card incorporates the driver's personal details as well as the truck owner's information. Drivers use their personalized cards to authenticate themselves to the system before driving the truck. The card remains inside the card reader throughout the trip and cannot be removed unless the truck's engine is turned off. Data obtained from the onboard computer and the external sensor is sampled, processed and sent to the smart card.

The monitoring system is much more reliable than the traditional logbook and tachograph because it protects encoded data against tampering and fraud due to the security conditions and the system's passwords encoded inside the smart card chip. Authorized individuals will have access to the stored data after having authenticated themselves to the system. Further advantages the system offers are its low price, its ease of use as well as its compact size and reliability. Accident causes and responsibilities will be easily and accurately determined, as well as the driver's behaviour and habits behind the wheel.

3.2 System Overview

The system is designed to enhance safety onboard commercial vehicles and on the roads. A lateral accelerometer is interfaced to the system to detect evasive maneuvers and drowsy conditions. The speed parameter is generated according to a preset pattern representing a real-time driving experiment and is stored in a Notepad file. A C++ software code simulates the onboard truck computer and interfaces to a micorcontroller of type PIC16F877 through the computer's parallel port connection as shown in figure 3.1. This same microcontroller manages all the different tasks and commands as well as the interface between the input data and the smart card reader, which stores the information on the smart card chip. Authorized persons equipped with specific codes will only have access to the recorded data, which cannot be modified under any circumstances. Figure 3.2 shows the connection between the Smart Card reader and the enforcing authority terminal.

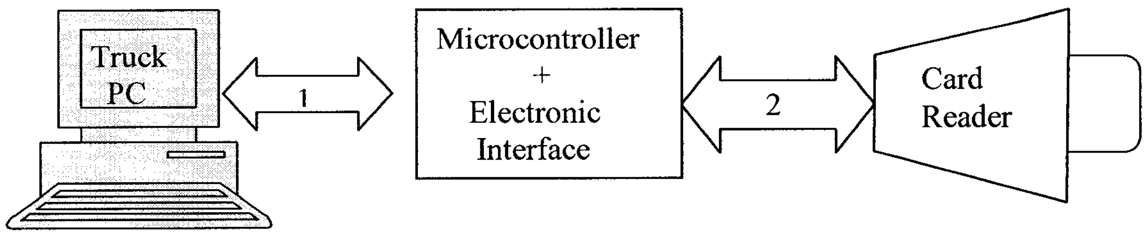


Figure 3.1: System setup for truck usage

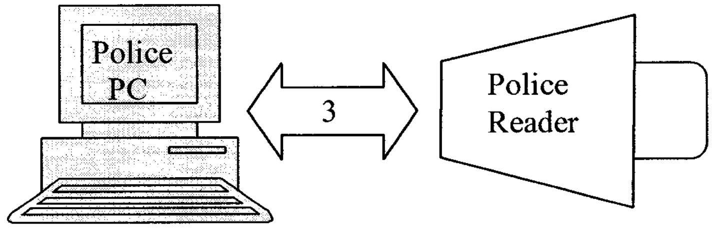


Figure 3.2: System setup for authority usage

3.2.1 The Recording Device

The recording device is a Smart Card called ACOS1. The COS, the smart card's Chip Operating System, is a sequence of instructions, permanently embedded in the ROM of the smart card. Like the familiar PC DOS or Windows Operating System, COS instructions are not dependent on any particular application, but are frequently used by most applications. The card has 8 KB of EEPROM memory with built-in security codes and can only be read via a reader of type ACR30 built specially for similar card types. The ACS Smart Card Reader/Writer (ACR30) has two interface tasks, the first being with the authorities computer and the second being between the microcontroller circuit and the smart card itself through a serial asynchronous interface (RS-232). The reader accepts incoming commands, carries out the specified function to the smart card and returns the requested data or status information according to the specifications of ISO7816-3.

The Smart Card contains two files; the first is an identification file that stores the driver's identification and personal information, and the second file is used for storing the recorded and processed data. Having only a maximum capacity of 255 records, the first file holds eight records whereas the second file holds 242 records made up each of 32 bytes in length. The remaining 5 records are left unused for future implementations.

3.2.2 The Data Processor

The Data Processor is the central processing and managing unit. It reads the speed every two minutes and generates the peak value and average data over five readings. Every two minutes interval, the measured value is stored in a specified register that is averaged over 5 readings and a peak value is generated for the calculated average. As an example, the lateral acceleration values, monitored every 2 seconds, are sorted according to their level of safety significance. Figure 3.3 illustrates qualitatively the divisions of the threshold values which are supposed to correspond to critical values. The limits may vary from driver to driver and from vehicle to vehicle. The data used for the purpose of this study is set according to a logical guess.

<table>
<tr><td colspan="2">Dangerous Maneuver</td><td colspan="2">Safe Region</td><td colspan="2">Drowsy Region</td><td colspan="2">Safe Region</td><td colspan="2">Dangerous Maneuver</td></tr>
<tr><td colspan="5">Vehicle turning left</td><td colspan="5">Vehicle turning right</td></tr>
</table>

Figure 3.3: Lateral acceleration value divisions.

The acceleration data being continuously analyzed every 2 seconds, triggers an alarm if the incoming signal value is in the drowsy region for a period exceeding 6 seconds and increments a counter if the dangerous maneuver region is detected. The counter is used to check the number of dangerous swerves the vehicle undergoes in a 2 minutes interval, which reflects the driver's driving condition and helps determine, in case of accidents or check points, his responsibility.

The real time is stored along with the two previous parameters in a specific location inside the microcontrollers EEPROM memory. When the programmed memory capacity of 32 bytes inside the microcontroller is reached, the stored data is securely transferred to the Smart Card according to a predefined transmission protocol, set by ISO7816-4, the ACS Company and the author. Storing the bytes in the microcontroller EEPROM memory prior to their transmission ensures a maximum card memory storage efficiency that is of 32 bytes per record. In addition, programmed routines in the PIC microcontroller program memory allow the processor to determine if the card is removed during operation of the vehicle, engaging an alarm and turning the truck off immediately. A further step in the system's security is the mutual authentification between the microcontroller and the card reader through a security code sent every time the card is being inserted into the terminal or data is being written onto the smart card.

3.3 The Monitored Data

The data monitored reflects the most significant and valuable road-safety related parameters. The driving time measures the permissible number of driving hours and is set according to the Canadian regulations issued by the provincial Ministries of

Transportation [16] which states that no driver is allowed to drive for a continuous period exceeding 13 hours and an on-duty period of 15 hours per work cycle, without having an 8 hour mandatory rest period. Excessive driving leads to excessive fatigue, which leads to drowsiness and an increase in the probability of accidents taking place. According to a study conducted in the USA by the National Transportation Safety Board [19], 52% out of a sample of commercial vehicle drivers admitted to falling asleep on the wheel and the estimates show that fatigue-related fatal accidents are estimated to be around 30%. The lateral acceleration replaced by a yaw rate sensor is intended to monitor the driver's alertness whereas the speed parameter is implemented in order to measure excess speeding, which is among the leading accident factors. A study conducted by Brooks in Australia [50] states that speed in heavy vehicles is more critical for truck safety than for any other vehicle in general because these vehicles operate closer to the physical limits and they have longer braking distances as well as a greater reaction time compared to cars. A software designed at CONCAVE [51] intended for the computers onboard authorities vehicles, shows graphically the driving pattern recorded by the system and a color quota is devised to highlight driving infractions according to their level of severity, to make it easier for authorities to analyze the driving behaviour and habits of drivers and take the necessary and appropriate measures.

3.4 Summary

The several parts of the system are introduced and the various connections between the data recorder, the smart card terminal, the sensor and the external computer are determined. The lateral accelerometer data collection is described with its different possible combinations and the countermeasures taken by the system are discussed. The processing and storage of the recorded information is explained briefly and justified whereas to their relevance in road safety.

CHAPTER 4

THE SMART CARD

4.1 Introduction

The smart card is a credit card sized plastic card embedded with a chip module containing a single silicon integrated circuit chip with a memory and a microprocessor. The chip module is a very fragile component that cannot be simply laminated to the surface of the card, therefore needing a sort of enclosure to protect it from the rough everyday life usage. The body of a smart card inherits its fundamental properties from its predecessors, the familiar embossed card. These cards, which still dominate the market in the credit card sector today, are simply personalized plastic cards with a variety of features like the name and number of the cardholder. Improved versions of these cards are provided with a magnetic strip that allows simple machine processing. The next step in the development of those cards consisted of implementing a microcontroller chip on the card, hence the name microcontroller unit card or MCU, which maintains the same physical properties of its predecessors [34,35,49].

The aim of using smart cards in commercial motor vehicles, is to minimize fraud attempts, make onboard recorded data more secure and accurate, reduce and facilitate paperwork for drivers, as well as help concerned authorities obtain reliable information in critical accidents or in commercial motor vehicle checkups.

This chapter discusses the characteristics and features of the smart card. The operating system is explained and the communication protocols are described, as well as the onboard-recorded data structure and the implemented security.

4.2 Smart Card Properties

The ACOS1 is a microprocessor based smart card with 8 KB of EEPROM memory. It is compliant with the ISO 7816-1/2/3 standards and therefore abides by a set of international regulations that define its physical and electrical properties.

4.2.1 The Physical Properties

The ISO 7816-1 and 7816-2 specify the physical structure of a smart card. The format differs slightly from one type to the other but in general it has the dimensions of 85.60mm x 54mm x 0.76mm, as shown in figure 4.1. Figure 4.2 shows that the card is made up of three elements; the plastic card as well as a printed circuit and an integrated circuit chip embedded on the card [8, 35]. The material employed for the manufacturing of the card body is the PVC (polyvinyl chloride) due to it's low cost, availability, easy to process feature and suitability for a wide range of applications [35, 8, 52].

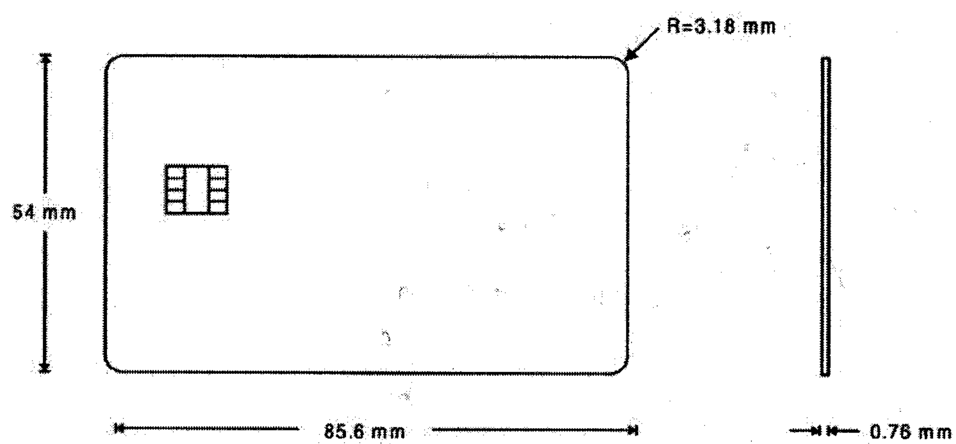


Figure 4.1: Smart Card dimensions [35]

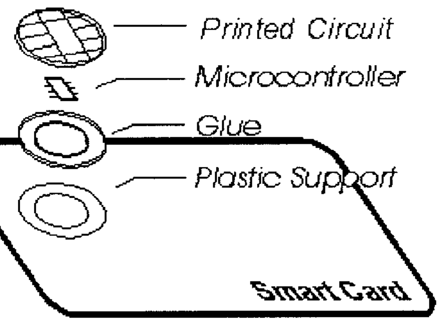


Figure 4.2: Smart Card components [8]

4.2.2 The Electrical Properties

The electrical properties of the smart card depend solely on the embedded chip since it's the only component of the card with an electrical circuit. As shown in figure 4.3, it interfaces with the outside world through six or eight contact areas defined according to ISO 7816-3. The nomenclature of the contacts is defined in figure 4.4.

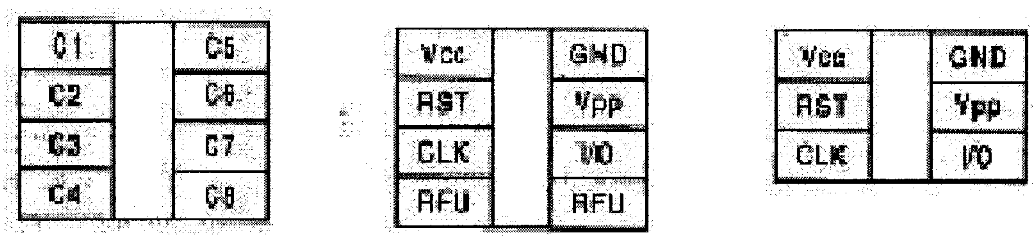


Figure 4.3: Electrical assignments and numbering of Smart Card contact areas according to ISO 7816-2 [35]

Contact	Designation	Function
C1	Vcc	Supply voltage
C2	RST	Reset input
C3	CLK	Clock input
C4	RFU	Reserved for future applications, presently not used
C5	GND	Ground (earth)
C6	Vpp	Programming voltage (generally no longer used)
C7	I/O	Serial communications input/output
C8	RFU	Reserved for future applications, presently not used

Figure 4.4: Contact designations and functions according to ISO 7816-2 [35]

The card obtains its power, a constant supply voltage of 5V DC and a current consumption of 50mA via its Vcc contact C1. Since the smart card does not have an internal clock generator, an external clock with a frequency ranging from 3.6864 to 4 MHz is therefore necessary to drive the logic IC and be a reference for the serial communication link [35, 49, 53]. The ACOS1 Smart Card, being an MCU based card, interfaces with its terminal device through six gold contact pads of type Siemens.

4.3 The Smart Card Operating System

The evolution of operating systems for smart cards has evolved along the years pushed by a steadily increasing market demand for individually customized solutions. The smart card's Chip Operating System, frequently referred to as COS, is a sequence of instructions permanently embedded in the ROM of the card. This feature leads to severe limitations on programming methods because no changes are possible once the microcontroller ROM has been manufactured and programmed. An advantage of ROM programming is that the operating system is very reliable and robust which is due primarily to error corrections that are extremely expensive and time consuming.

The COS has baseline functions that are common across all smart card products and they include:

- The management of interchanges between the card and the outside world along certain set protocols.
- Management of the files and data held in memory.
- Access control to information and functions, i.e. select file, read, write.
- Management of card security and the cryptographic algorithm procedures.

- Maintaining reliability, particularly in terms of data consistency, sequence interrupts, and recovering from an error.
- Management of various phases of the card's life cycle i.e. microchip fabrication, personalization, active file and end of file.

A typical price for a smart card ranges from 2$ to 10.00$ US. The ACOS1 smart card costs 6 $US and has an EEPROM capacity of 8K and according to the ACS company, manufacturer of the smart card, ACOS1 is specifically designed for highly secure payment applications such as network access control, electronic purse and multitude of other applications that stress high security which makes it suitable for safe and reliable data recording onboard commercial motor vehicles [35, 50, 53].

4.3.1 The Personalization Stage

The personalization stage of a card involves programming information related to the file structure, memory map and level of security. It is effective from the moment of termination of the manufacturing stage until the Personalization bit, an associated bit in the EEPROM, is set.

In order to personalize the ACOS1, a few steps done through the company's software and specified by the card manufacturer have to be followed. After powering up the card, the Issuer Code (IC) provided by the manufacturer must be submitted, and the personalization file (File ID = $FF02_H$) selected. This file contains features of the operating system that are divided into four bytes representing respectively, the Option Register, the Security Option Register, the Number of files required for the application and the Personalization bit. The first two bytes respectively specify banking options, which will not be implemented due to their irrelevance to the application and encryption options corresponding to the secret codes present in the application. The third byte allocates a number of data files in the File Data Area and the Personalization bit stored in the fourth byte serves as a fuse which once set, ends the personalization stage. Values of 00h, 00h and 02h corresponding to the Option Register, the Security Option register and the N_OF_FILE respectively are entered to specify that no options have been set in the first byte, no encryption option has been set and two files have been chosen for the application. A card reset must be performed following the previous steps in order for the smart card to accept the new values entered. The default IC code is submitted again in order to proceed with the personalization.

The next step consists of writing the definition blocks for the required user files contained in the File Management file (File ID = $FF04_H$). The file's record lengths as well as the read and write security attributes of the two files already implemented are set. The Security File (File ID= $FF03_H$) and all the keys and codes are then initialized. The Personalization File is invoked once again to initialize the Security Option Register and set the Personalization Bit that will lock the card and personalize the card according to the several features programmed. A card reset is done as a final step to set the card in the user stage [35, 53].

4.3.2 Memory Organization

The ROM, Read-Only-Memory, is programmed all at once during the manufacturing stage. It is constant for the rest of the chip life and stores the program code for the operating system. The EEPROM, Electrically Erasable Programmable Read Only

Memory, is the main part used for storing the recorded data. An 8 KB memory area is provided by the chip and is basically divided into Internal Data Memory and User Data Memory. The first sited is used for data configuration storage and is also used by the card operating system to manage functions. The User Data Memory, which stores the data manipulated during the application is allocated a memory space that is predefined according to the level of security encryption, the use of account data structures as well as the number of defined files during the personalization stage. The memory maps and size after the personalization stage differs from one application to the other. The application related to the dissertation is divided accordingly:

Memory space for User Data File = 7962 – N*6

Where N = number of User Data Files defined.

Hence Memory space =7962 – 2*6 = 7950 bytes

The size of a User Data File is calculated as:

Number of Records * Record length (bytes)

and is adjusted to the next higher multiple of 4 bytes, hence, the full User Data memory space is the sum of the individual file sizes. [35, 53]

4.3.3 File Structure and Management

Smart Cards have a complete hierarchical file management system with a symbolic hardware-independent addressing, meaning that all files are addressed by hexadecimal codes and all commands abide by the same type of communication. File management systems in Smart Cards have an object-oriented construction; thus all the file information is stored in the file itself, which is divided into two parts, as shown in figure 4.5. The first part, called the file header, contains information about the file layout, its structure and the access conditions. The second part stores all the variable data and is linked to the header by a pointer [34, 35, 52].

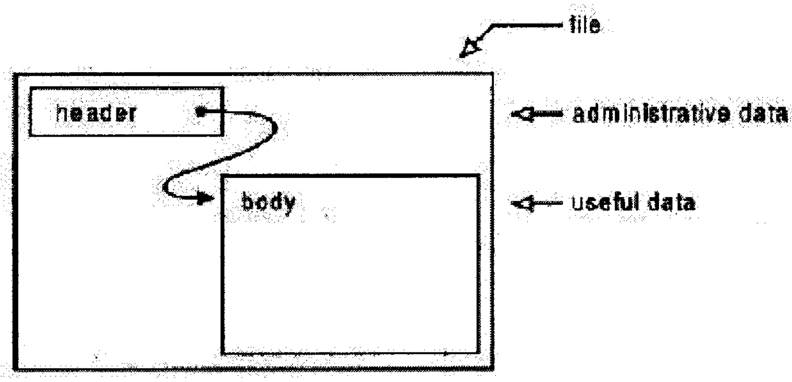


Figure 4.5: The internal structure of a file for a Smart Card file management system [35]

ACOS1 files are composed of data records, which are the smallest data units that can be individually addressed. Files are divided into two types of files, the Internal Data Files, which are stored in the Internal Data Memory and the User Data Files that lie in the User Data Memory. A summary of the internal and user files is shown in Table 4.1. The Memory Area represents the allocated space in the card for the files; the Internal File ID is the notation used to refer to the files; the Manufacturing Stage is the first step of the card programming; the Personalization Stage, following directly the Manufacturing Stage, allows the user to program the card chip according to the desired application; the User Stage designates the 'normal' operating mode of the card and is effective from the moment of termination of the Personalization Stage until the so-called Issuer Code is submitted to the card. A submission of the Issuer Code changes the operation mode to the so-called Issuer Mode. This privileged mode allows access to certain memory areas, which are otherwise not accessible. Figure 4.6 shows the several transitions between the four cited stages.

Table 4.1: Structure and organizations of ACOS1 files

		File Security Attributes			Record
Memory Area	**Internal File ID**	**Manufact-uring Stage**	**Personali-zation Stage**	**User Stage**	**Organization (Rec. x Bytes)**
MCU-ID File	FF 00_H	R: F.A W: N.A	R: F.A W: N.A	R: F.A W: N.A	2 x 8 bytes
Manufacturer File	FF 01_H	R: F.A W: IC	R: F.A W: N.A	R: F.A W: N.A	2 x 8 bytes
Personalization File	FF 02_H	R: F.A W: N.A	R: F.A W: N.A	R: N.A W: F.A	3 x 4 bytes
Security File	FF 03_H	R: F.A W: N.A	R: F.A W: N.A	R: N.A W: I.C	14 x 8 bytes
User File Management File	FF 04_H	R: F.A W: N.A	R: F.A W: N.A	R: F.A W: IC	Number of files (i.e. N_of_File) x 6 bytes
User File Data Area					
		User File Features			
Driver Identification File	F0 00_H	None	None	R: F.A W: IC	8 x 32 bytes
Recorded Data File	F0 01_H	None	None	R: IC W: IC	242 x 32 bytes

Where: F.A = Free Access

N.A = No Access

IC = Issuer Code

R = Read

W = Write

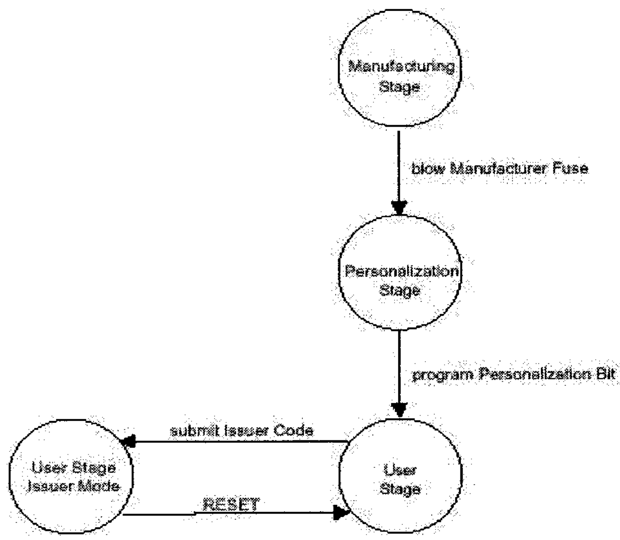


Figure 4.6: Diagram showing the possible transitions between the four stages [53]

The micro-controller unit (MCU) ID file and the Manufacturer file are both not accessed in the card personalization because they are programmed during the chip manufacturing stage. The two files contain data relevant to the operation of the embedded chip as well as the memory organization of the card. During the manufacturing stage, the manufacturer file, the IC code as well as the manufacturer fuse, the most significant byte of the first record of the Manufacturer file that indicates the termination of the manufacturing stage, are stored in the EEPROM. The Personalization file is set as defined previously in Section 4.3.1 and is shown in Table 4.2, where the two last records are not used in this application and therefore kept empty.

Table 4.2: Personalization File

Option Register	Security Option Register	N_OF_FILE	Personalization bit
00_H	00_H	02_H	80_H
Not Used	Not Used	Not Used	Not Used
Not Used	Not Used	Not Used	Not Used

The security file, FF 03_H stores the several codes and keys that protect the card's recorded information from illegal access and tampering as well as error counters for limiting the number of unsuccessful code presentations and authentications. The IC code, the PIN, the authentication keys and the application codes are entered sequentially in record 0 through 9.

Table 4.3: Security codes and keys

Record 0	Issuer Code IC	41 43 4F 53 54 45 53 54
1	PIN	31 32 33 34 35 36 37 38
2	Authentication Card Key Kc	41 55 54 48 43 41 52 44
3	Authentication Terminal Key	41 55 54 48 54 45 52 4D
4	Random number seed for RND_C =	41 55 54 48 43 41 52 44
5	Application Code AC1	41 55 54 48 43 41 52 31
6	Application Code AC2	41 55 54 48 43 41 52 32
7	Application Code AC3	41 55 54 48 43 41 52 33
8	Application Code AC4	41 55 54 48 43 41 52 34
9	Application Code AC5	41 55 54 48 43 41 52 35

PIN = Password set by the author that can be used for future security enhancements

Authentication Card Key Kc = Key code for authentication.

Authentication Terminal Key K_T = Key code for authentication.

Random number seed for RND_C = Random number generated by card.

Application Codes AC1, AC2, AC3, AC4, AC5 = Codes available to control the date stored in the data files.

The remaining 4 records of the security file are not implemented due to their irrelevance with the application in question.

As shown in Table 4.4, the User File Management file, FF 04_H is used to program the driver features related to the user files F0 00_H and F0 01_H and allocate EEPROM memory space for the file records as well as their read/write security conditions.

Table 4.4: User File FF 04_H Settings

Record Length	Number of Records	Read Security Attributes	Write Security Attributes	File Identifier Byte 5 / 6
20_H (max)	08_H	80_H	80_H	F0 00_H
20_H (max)	$F2_H$	80_H	80_H	F0 01_H

Each of the User File records can contain up to 32 bytes (i.e.20_H) of memory storage, which is also its maximum allowable capacity set by the manufacturer. The Security attributes set by the hexadecimal number '80', request that any read or write attempt from and to the card must be preceded by the submission of the Issuer Code IC. The User File F0 00_H contains 8 records of 32 bytes each that hold the drivers personal information, whereas the second User File F0 01_H, divided into 242 records of 32 bytes each, contains the recorded driving parameters [53].

4.4 Smart Card Data Transfer and Protocols

Digital data exchange between a card and its terminal always occurs via a single electric line of a serial channel. The card and the terminal take turns in sending data, while the other party acts as a recipient. The alternate data transmission and reception is called a half-duplex procedure. After a card is inserted into the terminal device, its contacts are first mechanically connected to those of the terminal. The five active contacts are then electrically enabled in the correct sequence followed by a power-on-reset that makes the card send an Answer to Reset (ATR) to the terminal. The terminal evaluates the ATR, which reports various parameters related to the card and data transfers and accordingly sends the first command. The card processes the received command and generates a response according to its validity, type and structure [34,35, 49, 51, 53]. The ATR protocol will be discussed in Section 4.4.2.

4.4.1 Asynchronous Half-Duplex T=0 Transmission Protocol

The communication between a Smart Card and its terminal device is done through a serial interface using the T=0 protocol where the T stands for Transmission protocol. There are 15 transmission protocols available for which basic functions are defined, but only two of them are predominantly used on an international level; the T=0 and T=1 protocols. The T=0 protocol will only be discussed in the dissertation due to its relevance with the project at hand.

The T=0 protocol is a byte-oriented protocol, which means that the smallest unit processed by the protocol is a single byte. Data is transferred asynchronously and each byte sent is provided with supplementary synchronization bits. As shown in figure 4.7, the low start bit is added to the beginning of each sent byte to indicate to the terminal the start of the transmission. A parity bit for error detection is added along with one or two stop bits at the end of each byte. The guard time is the time allocated to the stop bits in the T=0 protocol and is used by the sender and the receiver to prepare for the next byte transfer. The communication used for transmitting data between the smart card and the ACR30 is a serial asynchronous half-duplex communication, having one start bit, 8 bits of data, no parity bit and one high stop bit described as the guard time.

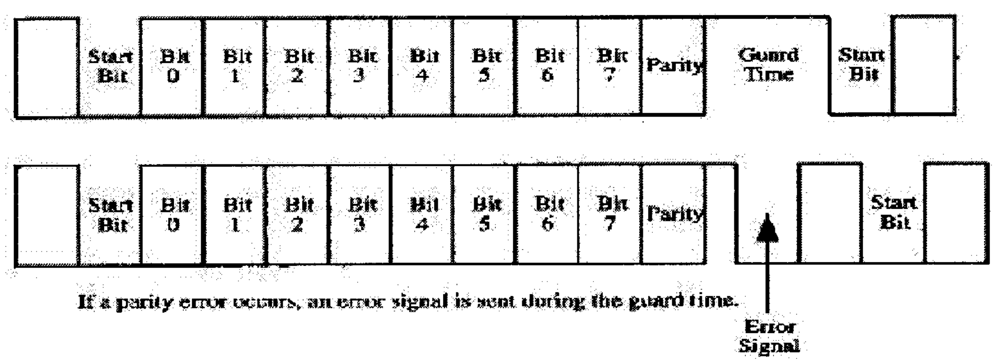


Figure 4.7: Reader-to-card byte transfer and error feedback loop [34]

Smart Cards operate with an external clock signal, which makes it impossible to determine the time interval for an individual data byte in absolute terms. It is therefore specified in terms of the applied clock that gives the number of clock pulses per bit interval. The nominal bit duration is defined as one etu (elementary time unit). Having a smart card terminal that operates at a frequency of 3.6864 MHz, a clock divider of 372 is used to produce an approximate baud rate of 9600 bits per second.

Due to its byte orientation, the T=0 protocol immediately request the retransmission of a byte in case a transfer error is detected. As soon as the recipient detects an error, it sets the I/O line to low for the duration of one etu, starting halfway through the first bit interval of the guard time of the faulty byte. This procedure indicates to the other party that the most recent byte must be retransmitted. The guard time separates the individual bytes during the transfer, which allows both communication parties more time to perform the transmission protocol's functions [34,35, 49].

The transmission of a byte of data using the ACR30 is done using only ASCII characters representing the hexadecimal digits '0' to 'F'. Each byte is split into its upper and lower half byte and is transmitted using the ASCII character representing the hex digit value and following the described procedure stated above [53]. The data transfer, following standards set by ISO 7816-4 and the T=0 protocol, is made of a data unit that contains a header consisting of a class byte, a command byte and three parameter bytes that may be followed by an optional data part. The data transfer structure is shown in figure 4.8.

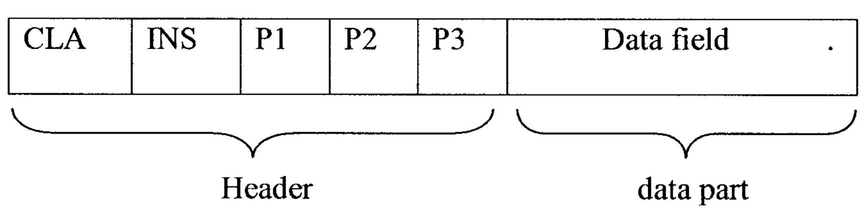


Figure 4.8: Structure of a command with T=0

4.4.2 The Answer To Reset (ATR)

The ATR is a string of characters returned by a smart card in response to a reset signal sent by the reader via the I/O line and contains various data relevant to the transmission protocol and the card. The ATR is at most 33 bytes and is always sent with a divider value of 372, in compliance with the ISO7816-3 standard. The IC chip is expected to respond during the period ranging from 400 to 40,000 clock cycles following the reset signal. The characters constituting the ATR are defined according to the ISO7816-3 standard.

The ATR sent by the ACOS1 card after receiving the reset is shown in figure 4.9 and is described in Table 4.5.

3B	BE	11 00 00	41 01 38 00 00 02 00 00 00 00 00 01 90 00	--

Figure 4.9: ATR string

Table 4.5: ATR character description

Designation	Value	Meaning	Remark
TS	$3B_H$	direct convention	
T0	BE_H	$Y1 = 1101_2 = B_H$ $K = 14_{10} = E_H$	TA1, TC1 and TD1 follow 14 historical characters
TA1	11_H	$FI = 0001_2 = 1_H$ $DI = 0001_2 = 1_H$	F=372 D=1
TC1	00_H	$N = 0$	No extra guard time
TD1	00_H	$0_H = 0000$ $T = 0$	no specific interface character follow T = 0 protocol is used
T1 … T14	$41_H \| 01_H \| 38_H \| 00_H \| 00_H \| 02_H \| 00_H \| 00_H \| 00_H \| 00_H \| 00_H \| 01_H \| 90_H \| 00_H$		Specific data of the Smart Card manufacturer
TCK	Not included		

The first byte 'TS', called the initial character, is the synchronization byte that determines the baud rate and the sense of logic. 'TS' is a mandatory ATR component and can only be sent in two allowable codes: $3B_H$ for the direct convention or $3F_H$ for the inverse convention.

The format character, T0 provides information necessary to interpret the remaining ATR characters. The most significant 4 bits use a bit map to indicate the presence or absence of TA1, TB1, TC1 and TD1. The remaining four lower bits give the number of bytes in the historical field and cannot exceed 15 bytes.

The interface characters TAi, TBi, TCi and TDi, which the presence in the ATR is optional, are used to select the protocol and parameters used for subsequent higher-level communication between the smart card and the reader. In the case of the example presented in Table 4.5, TD1 is set to 0, hence specifying that no interface characters are present in the ATR.

The global interface character TA1 is the divider otherwise known as the clock rate conversion factor F and is encoded in the upper nibble as FI. The bit rate adjustment factor D is in the lower byte nibble and is addressed as DI. The encoding of F and D specifies the transfer rates in accordance with the standard.

The bit interval of the ATR, called 'initial etu' is calculated as follows:

$$etu = 372/f \quad \text{(sec)} \qquad 4.1$$

where f is the clock frequency, equivalent to 3.6864 MHz in the application related to this dissertation. The transmission protocol that follows the ATR defines the bit interval independently of the ATR. This interval called 'work etu' is specified as follows:

$$\text{Work etu} = (1/D) \times (F/f) \qquad \text{(sec)} \qquad 4.2$$

Having a value of F=372 and D=1 will set the initial etu equal to the work etu. This initial etu is used during answer to reset and is replaced by the work etu during subsequent transmissions.

The global interface character TD1 contains in its higher nibble information regarding the presence or absence of subsequent interface characters whereas the lower nibble identifies the current transmission protocol.

The global interface character TC1 encodes an extra guard time designated as a hexadecimal number N.

The historical characters T1, T2...T14 are used to convey information related to the life cycle of the card.

The check character TCK, the last byte of the ATR is the XOR checksum of all the preceding bytes and is used in order to verify the correctness of the answer to reset transfer. The TCK byte might not be present under the T=0 transmission protocol [34,35, 49, 50, 53, 55].

4.4.3 The APDU Structure

The APDU, which stands for Application Protocol Data Unit, is used by the Smart Card and its recipient to communicate according to standards set by ISO7816-4 protocol. The APDU therefore consists of a set of commands issued by the terminal and a set of responses sent by the card that constitute the entire data exchange. Its purpose is to ensure the transfer of data, recorded by the PIC microcontroller, to the Smart Card.

4.4.3.1 APDU Command

The command APDU structure is composed of a header and a body, where the body may be of variable length or entirely absent depending on the command issued.

As shown in figure 4.10, the header consists of four elements, which are the class byte (CLA) that identifies applications and their specific commands, the instruction byte (INS) that encodes the actual commands, and two parameter bytes (P1, P2) that provide additional information about the selected command via the instruction byte. The APDU body follows the header and fulfills a double role. First, it specifies the length of the data section sent to the card as well as the expected data response from the card. Secondly, it contains the data associated with the command that is sent to the card. Lc represents the length of the data section sent to the card whereas Le represents the data section sent from the card back to the terminal device [34,35, 52, 53].

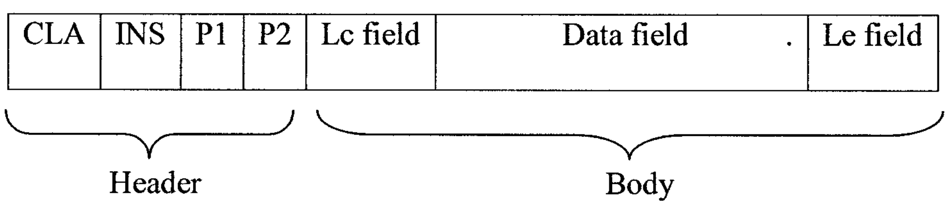


Figure 4.10: Structure of the APDU command

The PIC microcontroller, which communicates with the ACOS1 smart card through the reader/writer, encloses the APDU command along with a set a extra bytes that set the transmission of the desired commands. The communication protocol is shown in figure 4.11.

Start byte	Header	Instruction Code	Data Length	APDU COMMAND	Checksum	End byte
<STX>			LEN	From figure 4.10		<ETX>
02_H	01_H	$A0_H$				03_H

Figure 4.11: Command sent from PIC16F877 to ACR30

The <STX> and <ETX> hexadecimal bytes, which specify to the ACR30 the start and end of the byte reception, are the first and last bytes respectively sent in a certain command [52]. The header, which is always 01_H indicates the start of a command. It consists of only one byte and is completely different from the one stated in the APDU command structure. The instruction code $A0_H$ complies with the ISO 7816-4 and represents the CLA for the new command structure. LEN indicates the length of the total sent command and is defined as N+6, where N is the length of the enclosed APDU command. The checksum is a security byte and is obtained by XORing all the bytes preceding it except for the Start byte.

4.4.3.2 APDU Response

The APDU response is sent by the card in reply to an APDU command and consists of an optional body and a mandatory trailer as shown in figure 4.12.

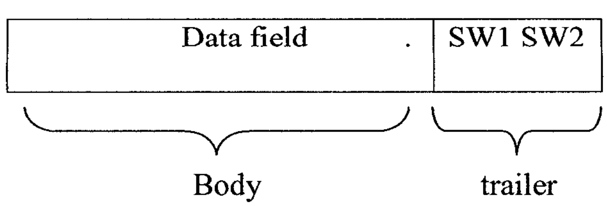


Figure 4.12: Structure of the APDU response

The body consists of the data field whose length is specified by the Le byte sent in APDU command. The trailer is always permanent in the response and is made up of two bytes SW1 and SW2, called the return bytes [34,35,52], which inform the sender about the status of the command sent.

The ACOS1 and the ACR30 return a simultaneous response having a specific type as shown in figure 4.13.

Start Byte	Header	SW1	SW2	Byte 1	…	…	Byte N	SW1	SW2	Checksum	End Byte
<STX>											<ETX>
02_H	01_H										03_H

ACOS1 Card Response

Figure 4.13: Response of the ACOS1 type card

The card response is incorporated inside the terminal's response and the bytes SW1 and SW2 return the status of the executed command for both the card and reader/writer device. Response status codes for the ACOS1 card and the ACR30 device are shown in appendix A. Byte 1 to byte N, in figure 4.13, represent the data field shown in figure 4.12 and the return bytes are the SW1 and SW2 values that precede the checksum part.

4.4.4 Implemented Commands and Responses

A set of instructions are used to communicate with the Smart Card. Commands, which are sent by the PIC microcontroller, according to the APDU structure defined previously are received by the ACR30 where they are processed and sent as commands to the Smart Card. The card then issues a response to inform the reader/writer about the success or failure of the transmitted command [52, 53]. Table 4.7 provides a sample communication between the PIC and ACR30.

Table 4.6: Commands and Responses used in the Application

1. *Power Up*

Command: 02_H 01 80 00 00 81 03_H

Response: ATR

2. *Submit Code*

Command: 02_H 01 A0 0E 80 20 07 00 08 41 43 4F 53 54 45 53 54 00 08 03_H

IC Code No   Issuer Code IC

Response: 02_H 01 90 00 00 02 90 00 03 03_H

3. *Select File*

Command: 02_H 01 A0 08 80 A4 00 00 02 File ID 00 CHK 03_H

Where File ID = F0 00 for personal driver file

F0 01 for recorded data storage

CHK = XOR Checksum

Response: 02_H 01 90 00 00 02 91 00 02 03_H

4. *Write Record*

Command: 02_H 01 A0 26 80 D2 [Rec. No] 00 20 DATA 00 CHK 03_H

Where Rec. No = Record Number

DATA = Byte 1 → Byte 32

Response: 02_H 01 90 00 00 02 90 00 03 03_H

5. *Power off*

Command: 02_H 01 80 00 00 81 03_H

Response: 02_H 01 90 00 00 91 03_H

Each command and response follows the APDU command and response described in figures 4.10 and 4.12 and the ATR response of the 'power up' command is the one shown in figure 4.8.

4.5 Security Architecture

The main characteristic of the Smart Card is that it offers a highly secure data storage environment. Although the ACOS1 offers DES (Data Encryption algorithm) calculations and mutual authentication as a part of its security architecture, a restricted set of security procedures are implemented in the application. The XOR checksum checks the authenticity of the issued commands and responses and the Issuer Code is used to access data in the restricted files of the Smart Card.

4.5.1 XOR Checksum

The XOR checksum is a very simple and secure way of detecting code errors during data transfer. It consists of performing a logical XOR operation on all bytes. Byte '1' is XOR-ed with byte '2'; the result of the operation is XOR-ed with byte '3' and so on until the last byte of the command is checked [34, 35]. The ACOS1 checksum is slightly different from the typical one due to its transmission protocol and the XOR checksum is separately performed on the odd and even bytes of the commands [52].

4.5.2 Security Codes

Secret codes in the card are used to restrict the access to data in the user data files and to certain commands provided by the card. A total of seven codes, defined in the security file FF 03_H, are present in the ACOS1 card and can be activated according to the level of security needed. For the purpose of the application, the Issuer Code IC, 'ACOSTEST' in ACSII code, provided by the manufacturer is the only activated code present on the card. This code must be presented to the card before any file access. A failure to do so or an attempt to present a false code will not allow the access to the information inside the chip. The PIN code is also implemented on the card but not in use currently. It is programmed to contain an eight-byte code ('12345678' in ASCII characters) and is reserved for future security enhancements.

4.6 Summary

Several features of the Smart Card developed to suite the application are described in detail. A brief discussion of the card properties is done and the formatting of the card otherwise known as the personalization stage is explained showing the several steps undergone during this process. The file structure and management are investigated highlighting the relevant options to the system and the two additional files, F0 00_H and F0 01_H used for the application purpose, are described in detail. The set of commands used in the data transfer to and from the Smart Card are explored as well as the security architecture implemented to prevent any fraud or tampering with the stored data.

CHAPTER 5

THE DATA PROCESSOR

5.1 Introduction

The data processor, which assumes the interface between the vehicle computer and the smart card terminal, is designed to read information related to two monitored road-safety parameters, namely the speed and the lateral acceleration. The received data is analyzed and processed along a certain predefined structure and sent sequentially to the smart card. In order to assume such a role, an electronic system that relies mainly on a microcontroller of type PIC16F877, is utilized. The PIC, or the central processing unit, is responsible for every instruction and command during the system's normal operation. Its task is to read the incoming data from the external sensor through an analogue to digital converter and from the vehicle computer through a parallel communication link. The data is then transformed into a suitable format. Extra needed parameters are added to the processed bytes, which are finally extracted to the smart card. The data extraction is accomplished via a serial communication protocol based on the APDU command structure described in detail in Chapter 4 of this document. The serial interface is controlled by a serial RS-232 driver/receiver IC of type MAX232 whereas the parallel communication link, managed by two 3-STATE Octal Bus Transceiver ICs of type DM74LS245, interface with an executable Visual C++ code onboard the host computer simulating the vehicle sensors.

This chapter discusses the various features that constitute the system. It describes the implemented hardware and the functions that define the developed software code used in the data acquisition, analysis and post-processing phases.

5.2 Hardware Design

With reference to figure 5.1, the hardware used in the application is divided into four parts: a digital electronic circuit for data processing, a Smart Card reader/writer, an external sensor and a computer terminal. The circuit, which constitutes the main part of the designed hardware includes a Microchip PIC microcontroller and contains two parallel buffers and a serial one. The electronic circuit is shown in figure 5.2. These serve as an interface with the external devices. The ACR30 Smart Card reader/writer is connected to the serial port whereas the computer is linked to the data processor through a parallel connection. The external sensor is linked through the A/D converter of the PIC chip at pin 2.

5.2.1 Electronic Circuit

Incorporated in the electronic circuit, The 'TRUCK IGNITION' connection, shown in the circuit layout, is set at either high or low depending on the status of the vehicle. A high signal represents a 'no driving' condition whereas a low signal indicates that the truck engine is running. A 555 timer is implemented between the microcontroller and the switch to reduce electric noise generated by mechanical contacts. The timer is set in monostable mode and generates a single-duration output pulse according to the trigger pulse applied at the input. The output timing ô is adjusted as follows:

$$\tau = 1.1 \times R \times .C \qquad 5.1$$

where the values of the external resistance R = 140 Ù and the capacitance C = 50ì F. The selected values result in an output time ô = 0.0077 seconds, which is experimentally justified (experiments undergone at CONCAVE research center, November 2002) and does not affect the normal functioning of the PIC microcontroller.

The ACR30 Smart Card reader/writer is connected to the circuit via the DB9-M connector shown in the circuit layout. This device, which requires a regulated voltage of 5V DC, 100mA, regulated power supply, draws its power from the extra cable joined to the main serial connector cable.

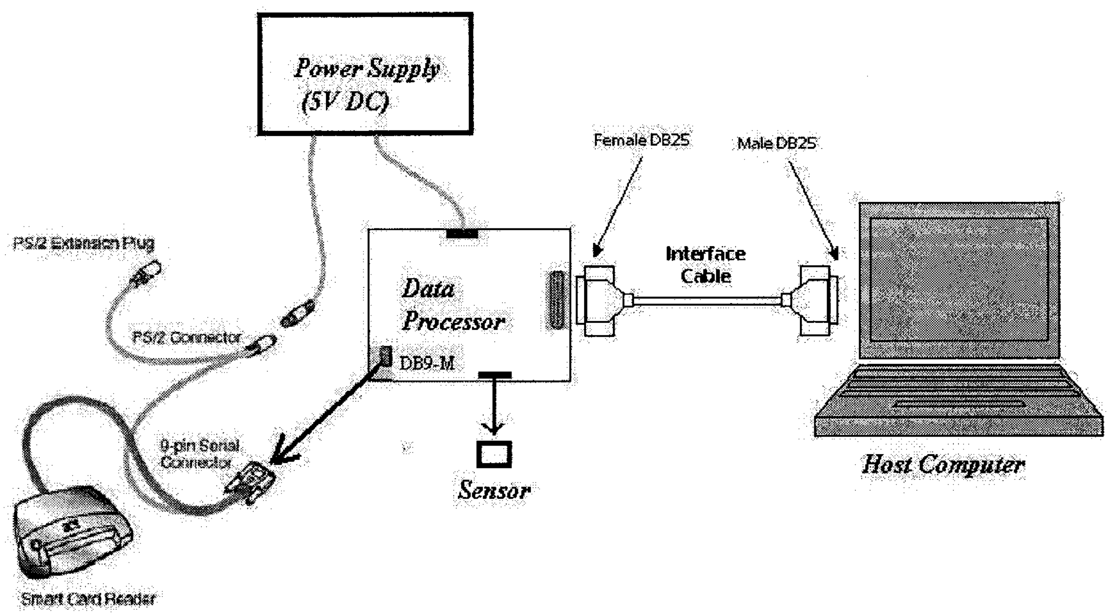


Figure 5.1: System overall Setup

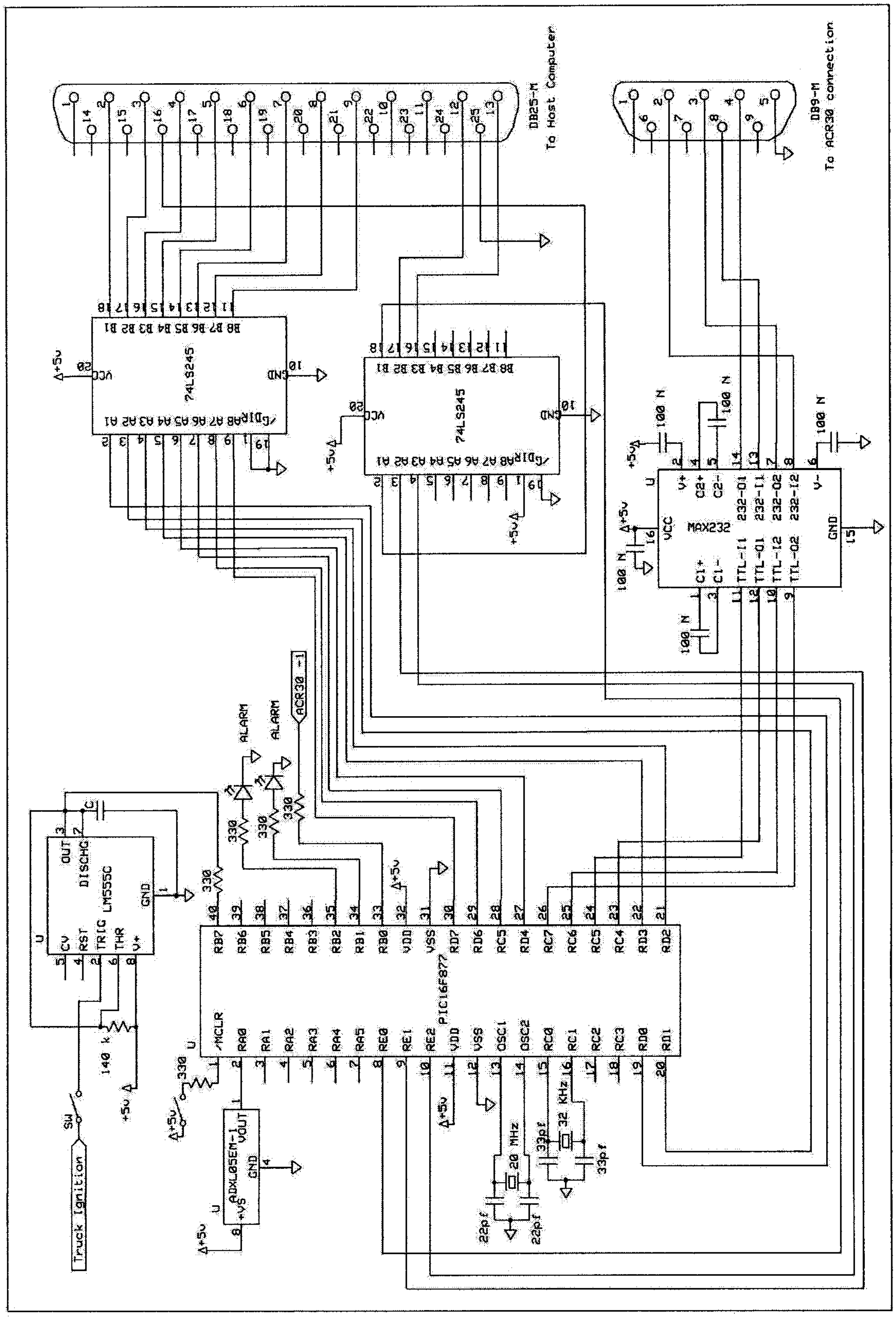


Figure 5.2: The electronic circuit representing the monitoring system

5.2.2 PIC16F877 Microcontroller

The PIC16F877 microcontroller is at the upper end of the mid-range series of microcontrollers developed by Microchip Inc. It is a 40-pin device that includes 8 Kbytes of Flash Program memory, 256 bytes of EEPROM data memory, 368 bytes of RAM and an 8-level hardware stack. The programming memory, divided into four parts of 2K each, is accessed using a 13bit wide program counter (PC). The data memory is partitioned into 4 banks extending from 00h to 7Fh, each holding 128 bytes of addressable memory that can be selected by the control bits, b5 and b6 in the STATUS register. The PIC contains an accumulator called Working Register (W) and other registers divided into Special Function Registers (SFR) and General Purpose Registers (GPR). The SFR, implemented as static RAM, are used by the CPU and peripheral modules to control the desired operations of the device. The STATUS register contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The microcontroller uses an external 20 MHz frequency crystal operating in HS (High Speed) mode and connected to the OSC1 and OSC2 pins. An additional 32 KHz crystal is used to operate Timer1, which is a 16-bit counter with a prescaler. The timer, programmed to overflow every 2 seconds, is used for the implementation of the calendar and the real time clock. Timer1's operation in Asynchronous Counter Mode and running with an external clock, will guarantee a valid read taken care of in hardware thus generating a precise sampling time of the monitored parameters.

5.2.3 Serial Communication

The data transfer between the ACR30 and the electronic circuit is done through serial communication. The hardware handshaking is accomplished using a five-wire connection following the RS-232 convention shown in Table 5.1.

Table 5.1: Serial Connection pins between Smart Card terminal and the microcontroller of the data processor

DB9 Pins	PIC16F877 Pins	ACR30	Function
2	RC7	RxD	Data transmitted to ACR30
3	RC6	TxD	Data transmitted from ACR30
4	RC5	DTR	Reset input signal. Allows performing hardware reset of the reader module by applying logic '1' (negative voltage) to the pin.
5	GND	GND	Reference voltage level for power supply and serial interface.
8	RC4	CTS	CTS (Clear to Send) indicates to the circuit whether the ACR30 is ready to receive the next command. Logic '0' (positive voltage according to the RS-232 convention) on this pin will allow the microcontroller to send a command.

The PIC microcontroller is set in UART (Universal Asynchronous Receiver Transmitter) mode to communicate with the ACR30 device. Half duplex communication is set in compatibility with the Smart Card terminal in order to operate in UART mode along with internal registers appropriate to the type of operation being undergone. The function 'SERIAL_SETUP', embedded in the program memory of the microcontroller, manages all the operation concerned registers. The Baud Rate, defined as the transmission rate of bits per sec is set according to:

$$BaudRate = \frac{F_{OSC}}{64 \times (X+1)} \qquad 5.3$$

where F_{osc} is the frequency of oscillation at which the chip is operating and X is an 8-bit datum stored in the SPBRG register as a decimal value. This register gives the appropriate sampling and shifting rates for the desired Baud Rate. For the 20 MHz frequency used and a Baud Rate of 9600 bits/sec, X is set to a decimal value of 129. During communication, the TXREG and RCREG, which are respectively, the serial transmit and receive accumulators utilized by the microcontroller, assume the byte-storage prior to a transmission or after a reception.

The microcontroller-ACR30 interfacing occurs through a RS-232 Driver/Receiver of type MAX232 [67]. This IC chip acts as a dual transceiver and translates the voltage at the PIC serial pins from +12V to 0V (logic 0) and –12V to +5V (logic 1).

5.2.4 Parallel Communication

The parallel port is utilized in the data processor system for data acquisition and communication between the implemented microcontroller and the truck-simulating computer. The parallel port, a D-Type 25 pin female connector, consists of three port addresses: data, status and control. The data port, composed of 8 data lines, is at address 0x0378, the corresponding status port, having 5 pins, is at 0x379 and the control port with 4 lines is at address 0x037A [55, 56, 57, 58]. Figure 5.3 shows the DB25 connector, where the letters “D”, “S”, and “C” represent respectively, data, status and control.

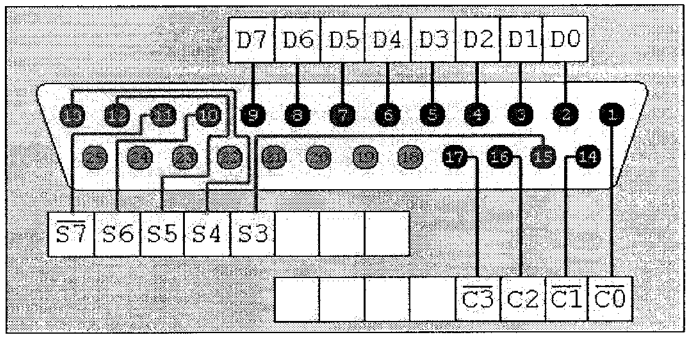


Figure 5.3: DB25 Parallel Port Connector

The PIC16F877 microcontroller uses two separate ports for its parallel communication and enables 8 bits of external data to be simultaneously and directly read into the processor. Port D operates as an 8-bit wide parallel slave port used for inputting data to the microprocessor whereas Port E, with pins RE1 and RE2 set as output pins and pin RE0 set as input, controls the communication between the microcontroller and its host computer. RE1 and RE2, connected respectively to pins S4 and S5 of the status port, select the parameter that is required by the system. A high input coming from C2 of the control port at RE0 indicates that the requested byte is at Port D and can be read by the microcontroller. The PIC to DB25 status and Control pin connections are summarized in Table 5.2 for reading the speed data from the PIC.

Table 5.2: Parallel communication connectors between the simulating computer and the microcontroller.

RE0 or C2	RE1 or S4	RE2 or S5	Function
-	Low	Low	No Data Requested
High	High	Low	Speed data is sent from PC to PIC
High	Low	High	Not in use
-	High	High	Never Occurs

The interface wire connection between the microcontroller and the DB25 pin connector of the computer is done through two octal bus transceiver IC chips. These IC chips of type DM74LS245 [68] are designed for asynchronous two-way communication between data buses and minimization of external timing requirements.

Table 5.3 describes the settings of the two ICs and shows pin 19 set as low for the two chips to keep them enabled at all time whereas pin 1, set as low and high, allows data to flow respectively into and out of the PIC microcontroller.

Table 5.3: DM74LS245 pin status

IC Chip	Enable $\bar{G}$ (pin 19)	Direction Control / DIR (pin 1)	Operation
Port D Interface Chip	L	L	PC to PIC
Port E Interface chip	L	H	PIC to PC

5.3 Software Design

The implemented software is developed with an assembly language specifically designed for the Microchip PIC microcontroller and uses a set of 35 instructions. The PIC instruction set is summarized in figure 5.4 and is presented in detail in the Appendix. The MPLAB Integrated Development Environment Software with pre-defined incorporated functions is used for development and simulation purposes before the assembly code is downloaded onto the chip.

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Figure 5.4: PIC16F877 Assembly Language Commands [55]

5.3.1 Assembly Code Description

The program starts by declaring, defining and initializing all variables and registers that will be used in later stages of the code as well as throughout the port settings. The serial communication registers are set and Timer1 interrupt is enabled with an overflow time of 2 seconds. Timer1 is a 16-bit timer/counter consisting of two 8-bit registers, which are readable and writable. It increments its two registers from 0000_H to

$FFFF_H$ and rolls over to 0000_H when an overflow occurs. In the project application, Timer1 is operating in asynchronous counter mode to allow the counter, which manages the internal programmed clock, to continue running during SLEEP mode. The Timer1 requires an external 32 kHz crystal to operate its built-in oscillator located between pins 15 and 16 of the microcontroller.

Table 5.4: PIC16F877 Port Settings

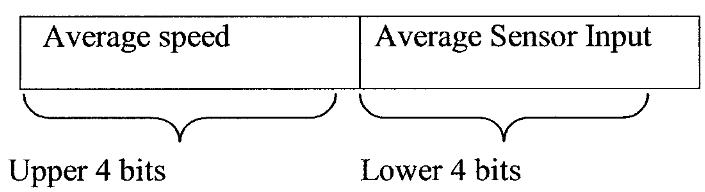
PORT	PIN SETTINGS	I/O TYPE
A	<0-5> Input	A/D
B	<1-3> Output, <0, 4-7> Input	External LED
C	<1, 3-6> Output, <0, 2, 7> Input	Serial Com.
D	<0-7> Input	Parallel Com.
E	<0> Input, <1-2> Output	Parallel Com.

Table 5.4, with reference to figure 5.2, shows the types of I/O ports used and their corresponding pins. The external lateral acceleration sensor is connected at Pin 0 of port A, where the signal is transformed from analogue to digital using the converter integrated in the chip and is read according to the format described in figure 3.2. In the main program function, the sampling time of 2 minutes for the speed and 2 seconds for the acceleration, the truck ignition condition and the card status are checked during each loop before the Timer1 overflow. Once all three conditions are met, i.e. overflow of the time counter, truck in motion and verification that the Smart Card is in the reader and has been authenticated, speed data is read at Port D and the accumulator storing the sensor

analyzed data is also read. After five samplings, the average and peak value for both parameters are computed, processed and stored in the EEPROM memory. The processing stage consists of an optimization phase involving the formatting of the computed bytes of data in order to maximize the storage capacity of the Smart Card.

The speed and the lateral acceleration sensor parameter average value results are in the range of the hexadecimal values of 0_H to F_H and 0_H to E_H respectively, defined for speed as 0_H = 0km/h and incrementing by 10 km/h for every bit increase up to F_H = 160km/h. The lateral acceleration is defined in terms of number of dangerous maneuvers as described in figure 3.2 of section 3.2.2. The sensor with a range of 0 to 5V DC and a sensitivity of 80 mV/g is calibrated to 2.75V or digital 80_H at rest. For simulation purposes, fictitious values are set in the PIC program. The drowsy region is delimited by the two values $7A_H$ and 85_H, where an alarm is triggered if the sensor value remains in this region for a period exceeding 6 seconds while the truck is in motion. Dangerous maneuvers are detected when the sensor value is either above 90_H or below 70_H and a counter is incremented every time a value is encountered in this region. The counter limit is set to E_H = 15 dangerous maneuvers and any additional infraction is no longer measured. Sensor values outside the specified ranges are considered to be normal driving conditions and thus not recorded.

Both values are combined into one byte as follows:



The peak, ranging from 0_H to F_H, values for both parameters are stored also in 1 byte as follows:

Peak Speed	Peak Sensor Input
Upper 4 bits	Lower 4 bits

Preceding the storage of the two mentioned average and peak values, the date and time bytes, which are also first reformatted to maximize memory usage in the Smart Card, are stored in the pre-allocated memory space.

Month-Year byte:

Month	Year
Upper 4 bits	Lower 4 bits

Where date range is defined from hexadecimal 0_H = January to B_H = December.

Day byte:

000	Day
Upper 3 bits	Lower 5 bits

Time byte:

Hours	Minutes
Upper 5 bits	Lower 3 bits

The time byte is always sent accompanying the parameters average and peak bytes whereas the day and the month-year bytes are sent as the first two bytes of a new card with values according to the calendar incorporated in the program code which tracks the local real-time date. Every second, the card insertion status and truck condition are verified using the function 'CARD_IGNITION'. Normal operation of the truck consists of having the correct card in the terminal during driving time. Any failure to abide by the set conditions or any attempt to retrieve the card while driving will automatically set an alarm and the system will go into off-duty mode. The off-duty mode is activated when the truck is normally shut down prior to withdrawing the card. A counter is incremented every time the data processor is supposed to read the parallel port incoming data during normal on-duty operation using the 'off_counter_function'. When the capacity of 32 bytes stored in the EEPROM is reached, the bytes are sent to the card. Several functions precede the writing process and a set of APDU commands is used to communicate with the Smart Card terminal. The 'POWER_ON' function powers up the ACR30 followed by the 'SUBMIT_CODE' function, which submits the IC code to the card. A response from the reader/writer captured by the command 'RESPONSE' analyses the incoming string of bytes and accordingly sets an error register in true or false state. In case of an erroneous code, the system sets an alarm on and goes immediately into off-duty state. On the other hand, having a positive response allows the continuation of the process and the 'SELECT_FILE' function followed by the 'WRITE' function selects the file F0 01_H and writes to the correct record, the 32 bytes of data. The 'POWER_OFF' command turns off the ACR30 and the system goes back to the main loop waiting for the next cycle to be triggered.

The APDU functions responsible for the transfer of data from the PIC16F877 to the Smart Card terminal use the serial asynchronous half-duplex communication along with a transmission protocol specially set for the ACR30. These mentioned functions are summarized in the flow charts of figures 5.6 to 5.10. The numbers in square brackets correspond to software functions listed in Table 5.5. References for the assembly code development are [57 to 66].

5.3.2 Implemented Functions

A set of 32 functions as listed in Table 5.5, are embedded in the microchip's program memory, composes the majority of the assembly code and is responsible for all its activities.

Table 5.5: Code Developed Functions

Function Number	Function Name	Description
1	AUTHENTICATE	Check authenticity of smart card
2	AVERAGE	Obtain average value for five consecutive readings
3	Calendar	Acts as Clock and calendar
4	Card_Ignition	Regulation management.
5	CHECKSUM	XOR APDU bytes
6	CHECK_BYTE_COUNTER	Check if the 32-byte capacity is reached in the EEPROM and send the data to the card if so
7	DATA_SEND	Retrieve data stored in PIC EEPROM and send to card
8	DAY_DATE	Format and store the day byte
9	DELAY	Delay function
10	EEPROM_READ	Retrieve data from EEPROM

11	EEPROM_WRITE	Store data in EEPROM
12	Get_file_number	Get current file number stored in EEPROM
13	Get_record_number	Get current record number stored in EEPROM
14	INCREMENT_FILE	Increment record number for next writing
15	Main	Main loop controlling the timing and sampling of data
16	MONTH_DATE	Format and store the month and year byte
17	Off_counter	Count off-duty time
18	PARALLEL_COM	Receive a byte from parallel port
19	PEAKS	Obtain the peak value of five consecutive readings
20	POWER_OFF	APDU command to turn off the ACR30
21	POWER_ON	APDU command to power-up the ACR30
22	RESPONSE	Check correctness of APDU commands
23	SELECT_FILE	APDU command to select file
24	SEND_OFF_DATA	Write to EEPROM off-duty time
25	SENSOR_INPUT	Reads the sensor input at Port A, Process signal and takes measures according to the result obtained.
26	SERIAL_REC	Receive a byte serially
27	SERIAL_SETUP	Set up all registers related to serial communication
28	SERIAL_TRANS	Send a byte serially
29	SUBMIT_CODE	APDU command to submit IC code to the Smart Card
30	TIME_DATE	Format and store the time byte
31	TRANSMISSION_PROTOCOL	Transmission of bytes according to ACR30 communication protocol
32	WRITE	APDU command to write data to a record

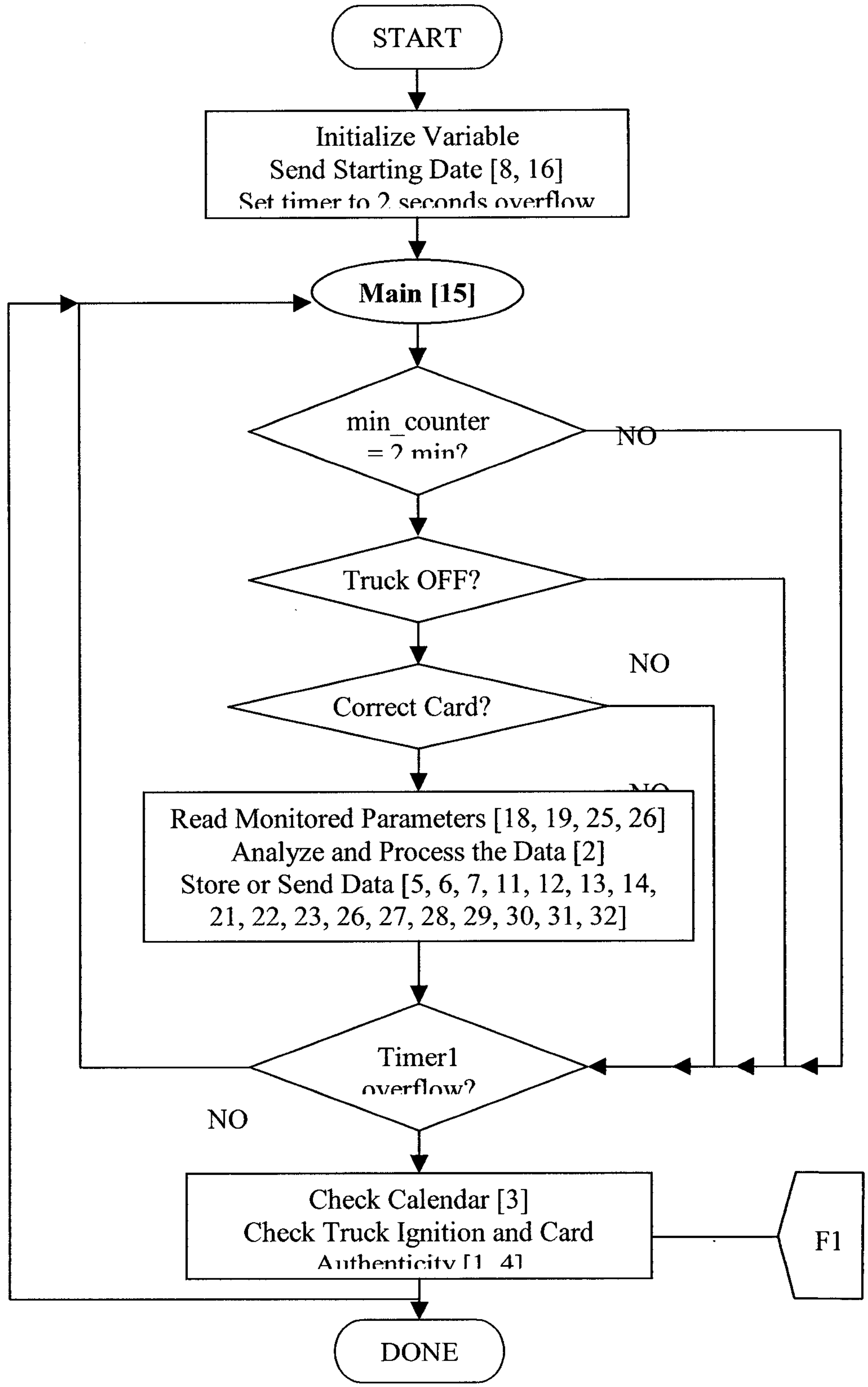


Figure 5.5: MAIN PROGRAM LOOP

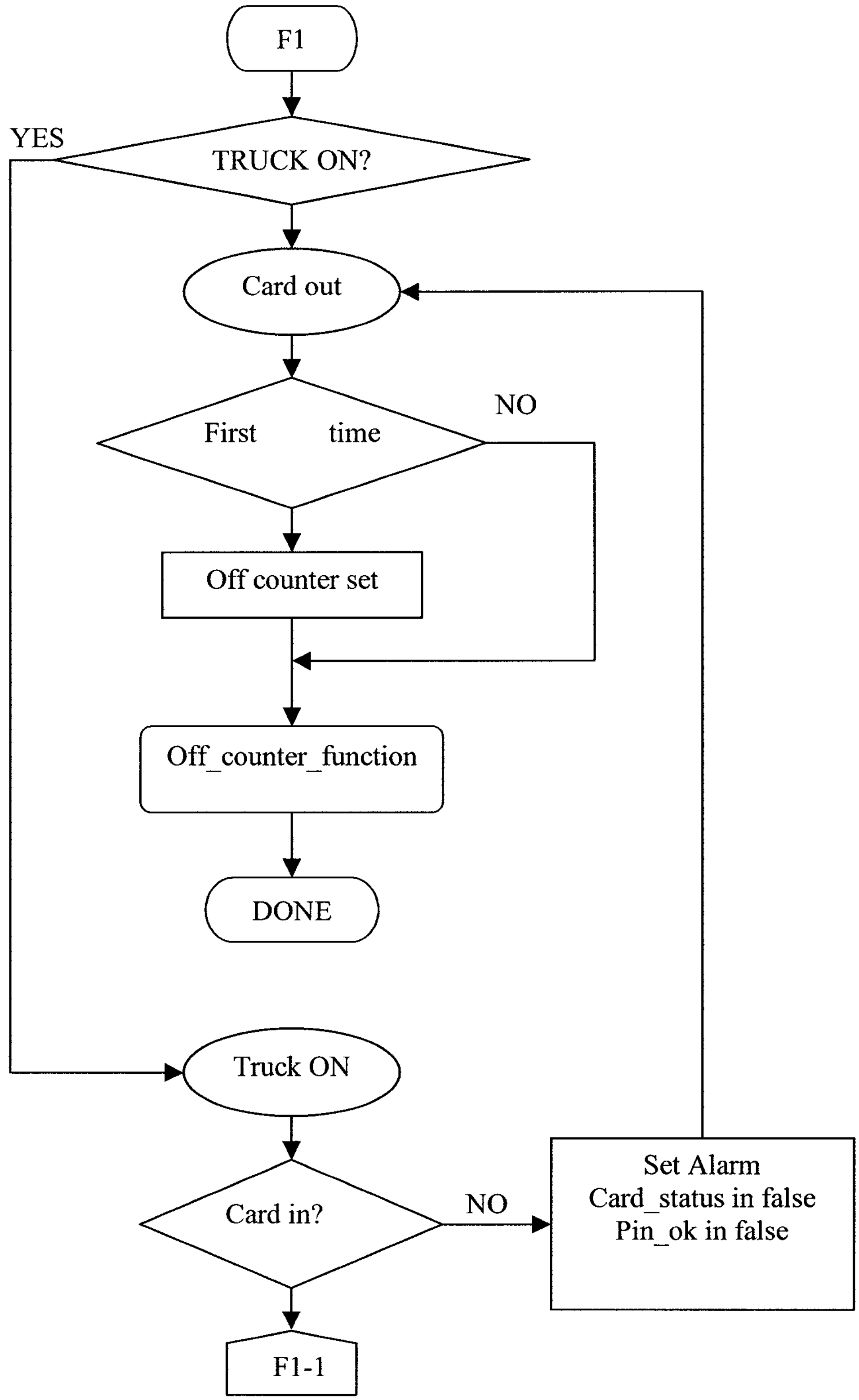


Figure 5.6: TRUCK_IGNITION & CARD_INSERTION VERIFICATION

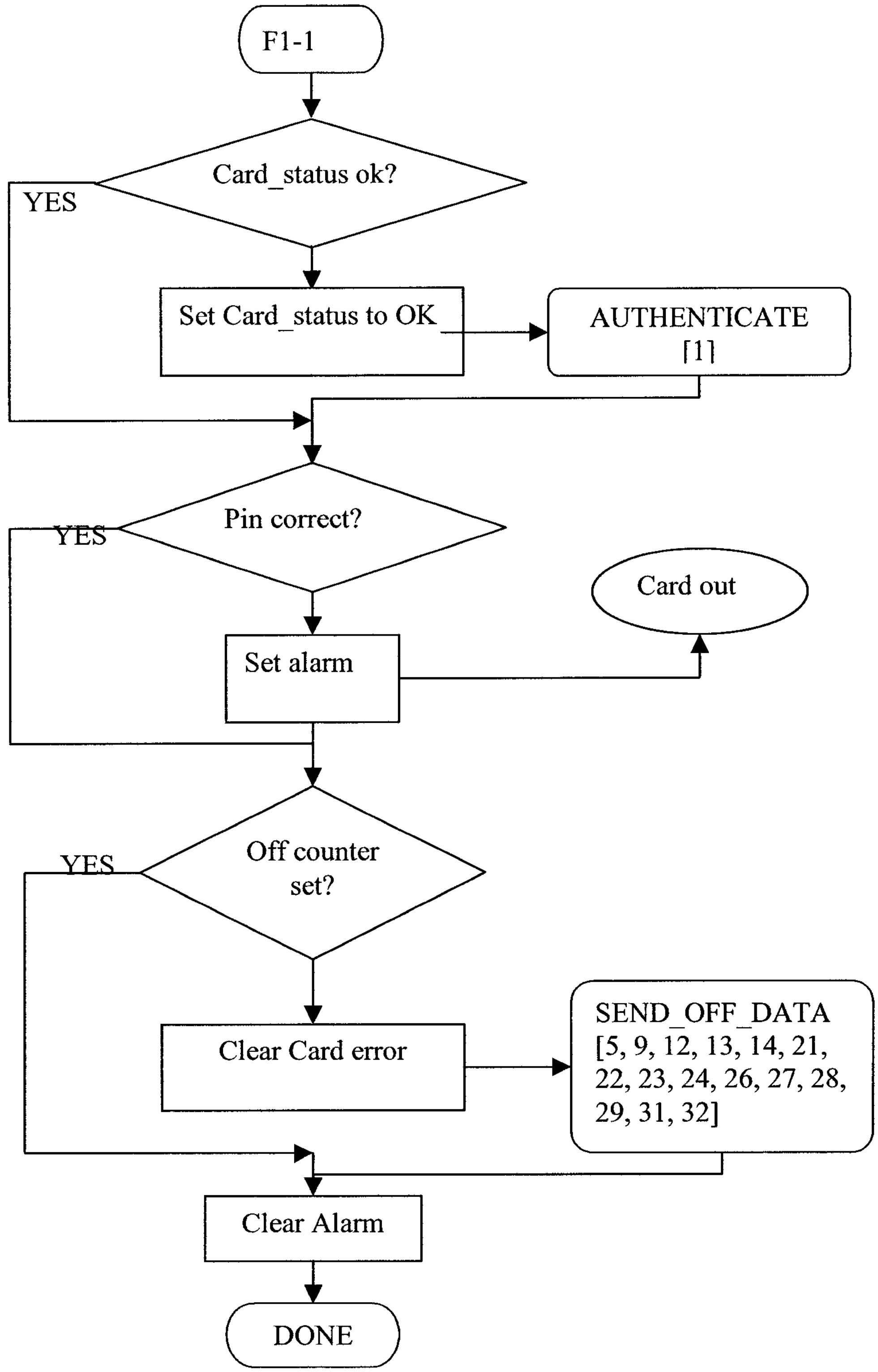


Figure 5.6: TRUCK_IGNITION & CARD_INSERTION VERIFICATION (continued)

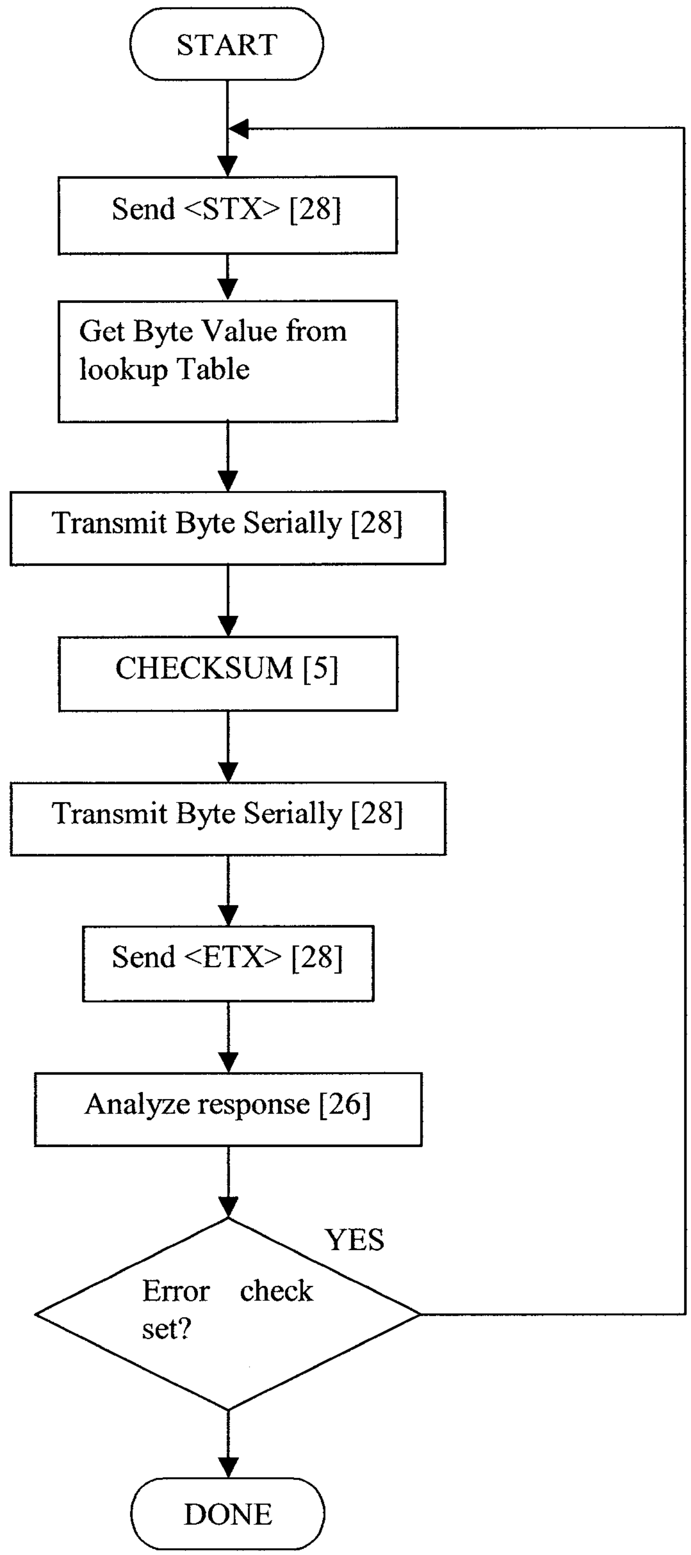


Figure 5.7: APDU Command

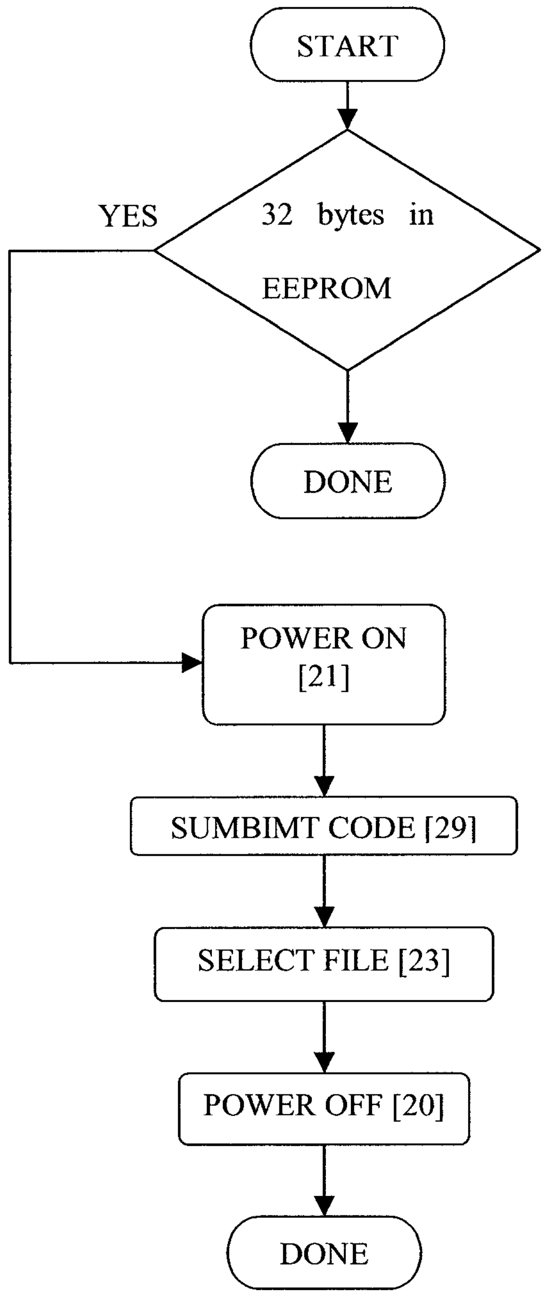


Figure 5.8: Sending data to Smart Card

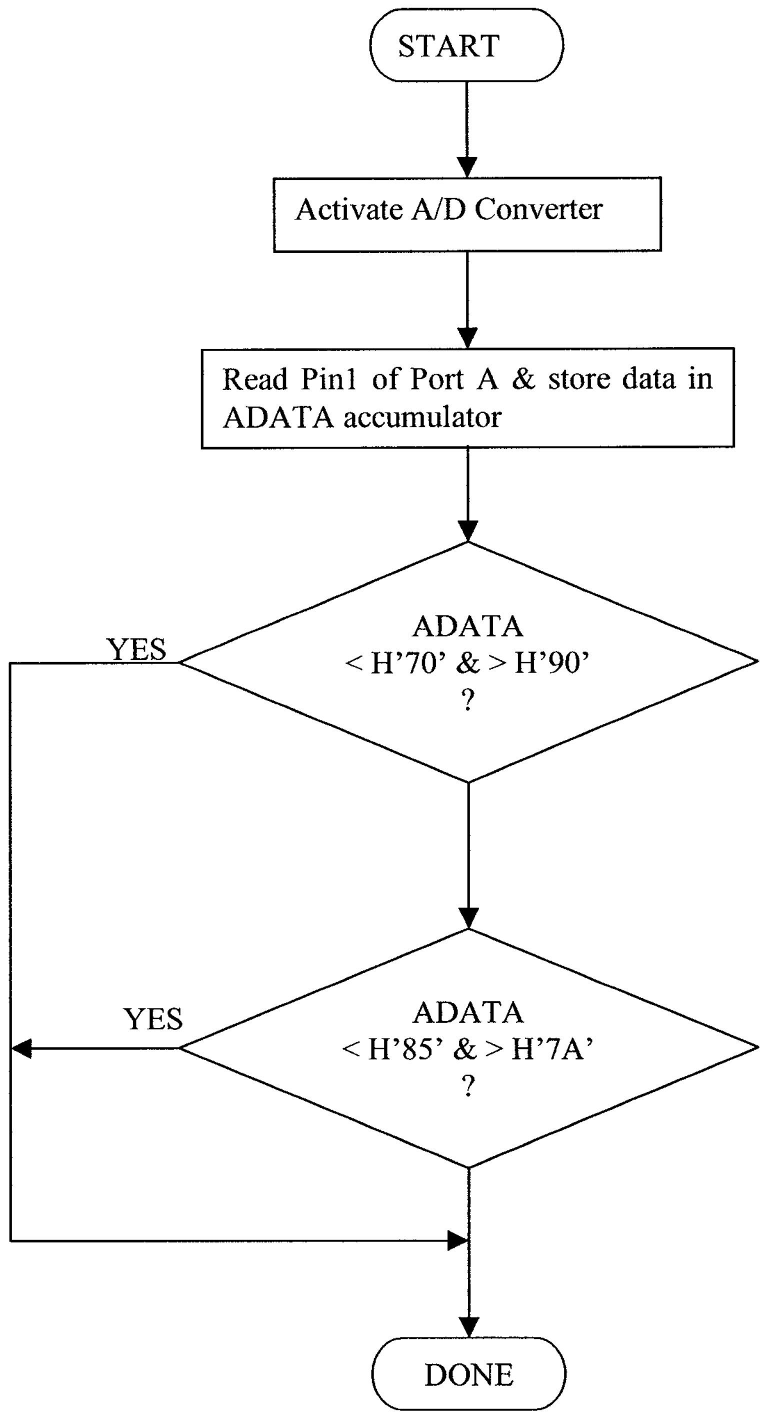


Figure 5.9: SENSOR_INPUT function: External sensor input processing

5.3.3 Memory Organization

There are three memory blocks in the PIC16F877 microcontroller: The program memory, the data memory and the EEPROM. The program memory holds the application code, which accounts approximately for 1.25 Kbytes of the 8 Kbytes Flash programmable memory and is stored between addresses 0x0200 to 0x06E0, thus occupying uniquely its first page. The data memory, partitioned into 4 banks as previously stated represents the RAM of the microprocessor and holds the special and general function registers (SFR, GFR). The registers defined for the system application are stored in Bank 1 in 96 bytes reserved for the General Purpose Registers (GPR) and can be accessed via direct addressing while indirect addressing is used for retrieving data from lookup tables. This indirect addressing mode is employed where the data memory address in the instruction is not fixed. The File Select Register (FSR), an SFR register, is used as a pointer to the data memory location that is to be read or written to. This type of accessing memory locations is integrated in the APDU functions of the system as a way of retrieving the command-related bytes located in lookup tables.

The Data EEPROM is a non-volatile memory that retains its contents when power is cut-off. The stored data is indirectly addressed through a set of Special Purpose Registers. EEPROM reading and writing are predefined in the PIC16F877 data sheet as shown in figure 5.10 and the microcontroller uses 32 bytes of memory to store data bytes, intended to be sent to the card's records. Six extra locations are also reserved for the file number, record numbers and date information in the EEPROM memory.

EEPROM DATA READ

```
BSF     STATUS, RP1     ;
BCF     STATUS, RP0     ;Bank 2
MOVF    ADDR, W         ;Write address
MOVWF   EEADR           ;to read from
BSF     STATUS, RP0     ;Bank 3
BCF     EECON1, EEPGD   ;Point to Data memory
BSF     EECON1, RD      ;Start read operation
BCF     STATUS, RP0     ;Bank 2
MOVF    EEDATA, W       ;W = EEDATA
```

EEPROM DATA WRITE

```
BSF     STATUS, RP1     ;
BSF     STATUS, RP0     ;Bank 3
BTFSC   EECON1, WR      ;Wait for
GOTO    $-1             ;write to finish
BCF     STATUS, RP0     ;Bank 2
MOVF    ADDR, W         ;Address to
MOVWF   EEADR           ;write to
MOVF    VALUE, W        ;Data to
MOVWF   EEDATA          ;write
BSF     STATUS, RP0     ;Bank 3
BCF     EECON1, EEPGD   ;Point to Data memory
BSF     EECON1, WREN    ;Enable writes
                        ;Only disable interrupts
BCF     INTCON, GIE     ;if already enabled,
                        ;otherwise discard
MOVLW   0x55            ;Write 55h to
MOVWF   EECON2          ;EECON2
MOVLW   0xAA            ;Write AAh to
MOVWF   EECON2          ;EECON2
BSF     EECON1, WR      ;Start write operation
                        ;Only enable interrupts
BSF     INTCON, GIE     ;if using interrupts,
                        ;otherwise discard
BCF     EECON1, WREN    ;Disable writes
```

Figure 5.10: EEPROM Read/Write Code [55]

5.4 External Software Interface

The data processor receives its data from a host computer that simulates the behaviour of a commercial vehicle on the road. The truck speed is stored in a 'txt' file and is accessed through a Visual C++ [69] code designed specifically by the author to interface with the microcontroller. The program reads the parameter values and transmits them to the computer's parallel port according to the set communication protocol defined in section 5.2.4. Figure 5.11 shows the program flow chart of the host computer for the transmission of the simulated vehicle speed to the PIC.

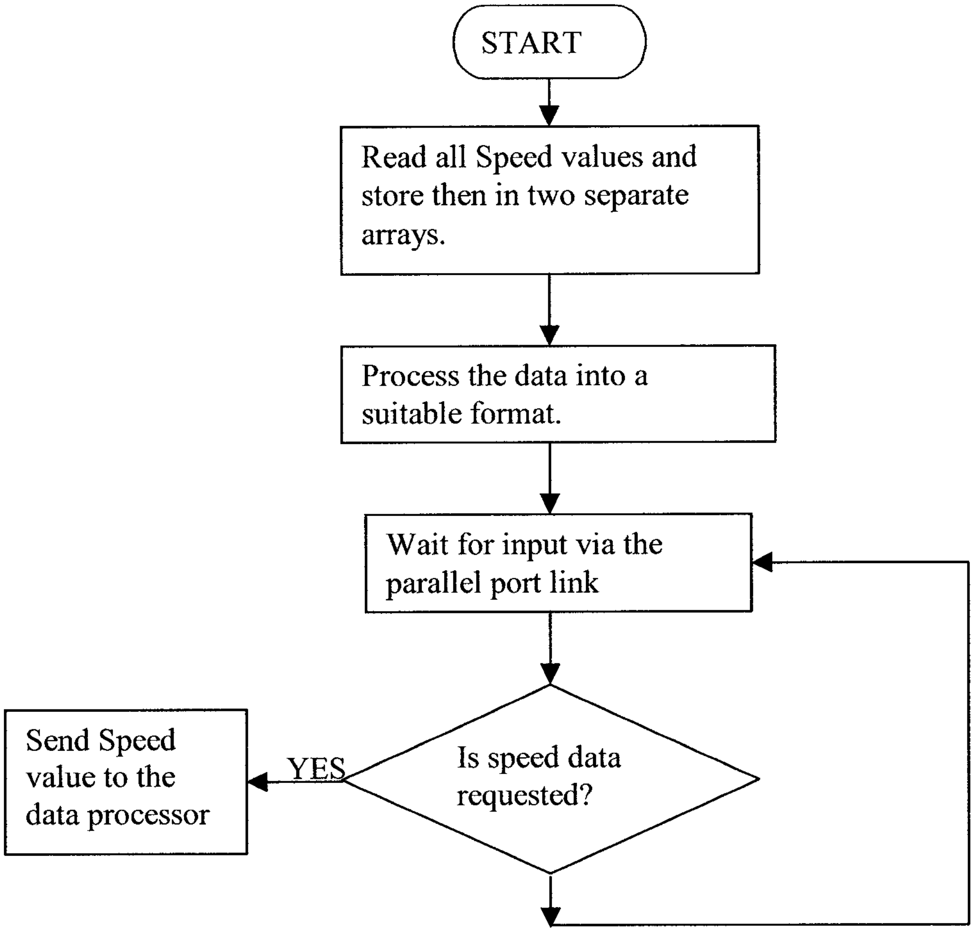


Figure 5.11: Flow Chart representing the speed simulation code

5.5 Simulation Results

Simulations for several driving conditions are performed using the monitoring system in order to show the several conditions that can be encountered in real life. The results are demonstrated using Visual C++ developed in section 5.4 and the computer-based software [51] designed specifically to read the smart card recorded data through an interface with the ACR30 terminal. Several driving sequences are simulated using the monitoring system and the results are shown in figures 5.12 to 5.14.

Figure 5.12 shows a random driving example. The different bars represent the calculated average speed over a 10-minute period. The text box shown in the figure is obtained by clicking on any bar. It contains information related to the driving time, the average speed and lateral acceleration counts, the peak speed and lateral acceleration counts, the date and time the data was recorded. The dashed horizontal line indicates the speed limit of 100 km/h to help enforcing authorities see clearly the graph.

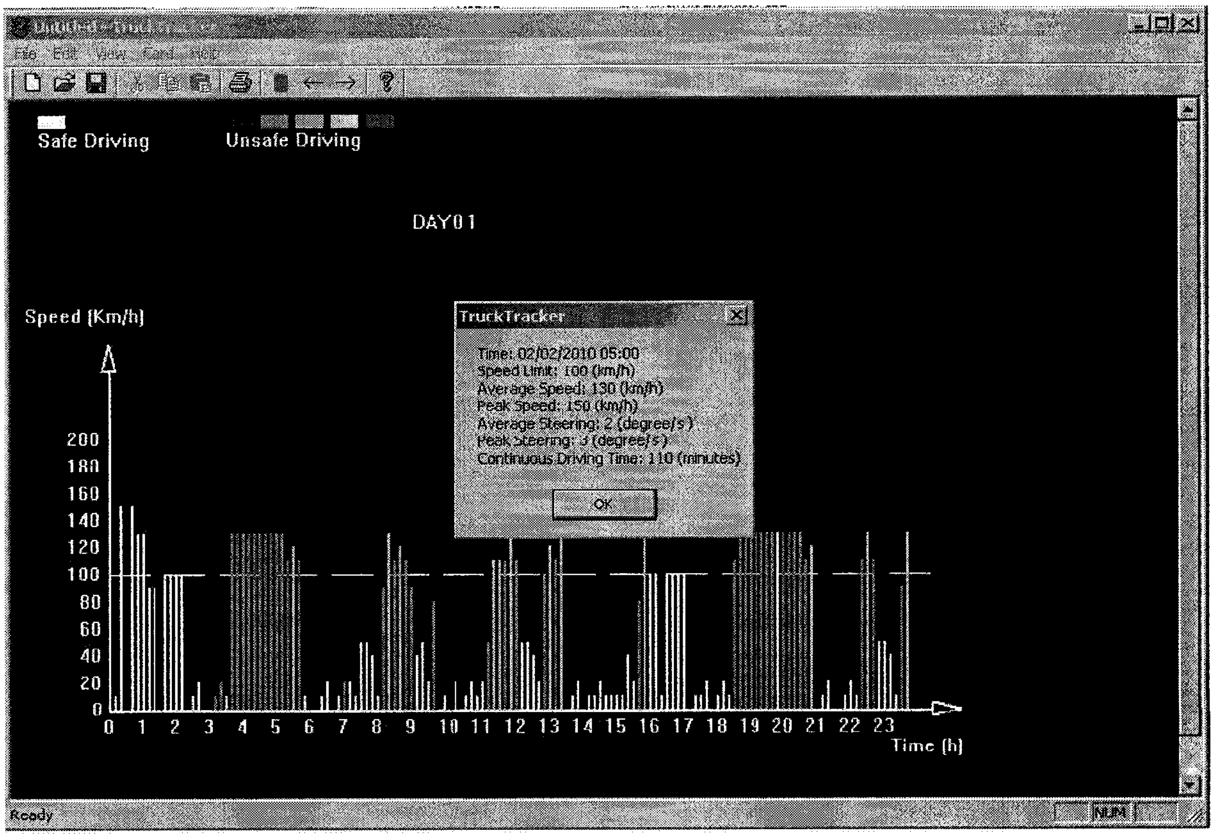


Figure 5.12: Random driving example

Figure 5.13 illustrates an off-duty period showing a speed average value of zero over a certain period of time. This state occurs in two conditions, the first being a normal rest time with the truck turned off, and the second is when the driver removes the card from the reader while driving. This action triggers an alarm inside the cabin, which does not appear on any record.

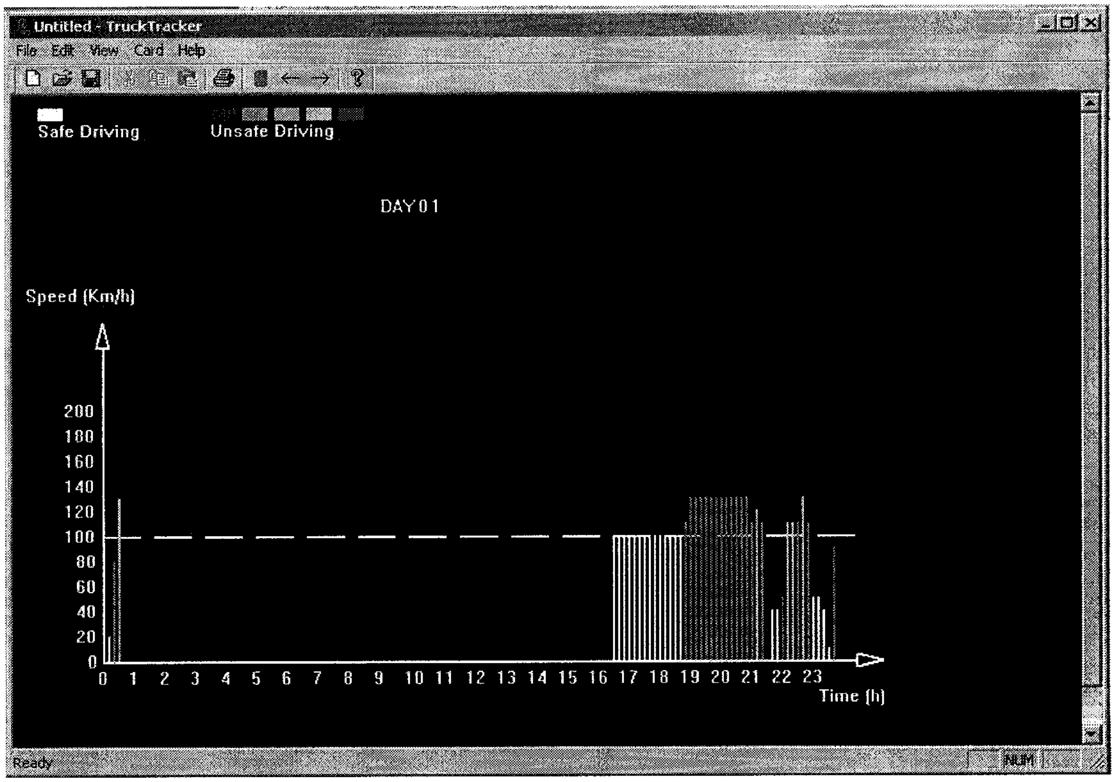


Figure 5.13: Off-duty period simulation

Figure 5.14 shows the contents of file F0 00_H that represents the profile of the driver using the card.

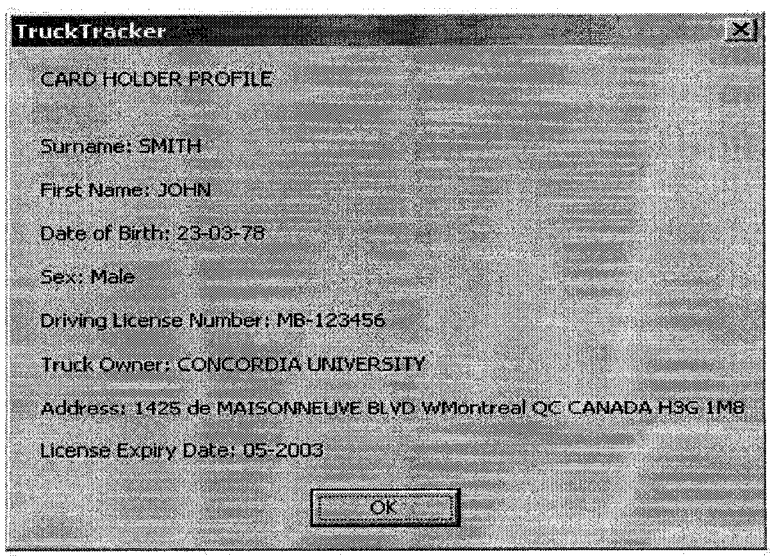


Figure 5.14: Card user profile

5.6 Cost Analysis

A brief summary of the components employed in the design of the entire system including the smart cards and the terminal device used to access the cards as well as the components of the electronic circuit are presented in Table 5.6. The total approximate cost to implement such a system is 200$ excluding the external sensor.

Table 5.6: Components price list

Component Name	**Unit Price in $ CDN**	**Number of Units**
PIC16F877	**10**	**1**
MAX232	**1.5**	**1**
DM74LS245	**1.5**	**2**
LM555CN	**1**	**1**
Lateral Accelerometer ADXL05EM-1	**150**	**1**
Resistors	**3.49 for 20 units**	**8**
Capacitors	**3.49 for 10 units**	**11**
LED	**3.49 for 10 units**	**2**
DB-9 Connector	**3.49**	**1 Male**
DB-25 Connector	**3.49**	**1 Male**
DB-25 Cable Connector	**10**	**1**
ACR30	**50**	**2**
ACOS1 Card	**10**	**12**

5.7 Summary

The data processor is designed and developed to monitor the driving behavior of truck drivers. Three major parts constitute the entire system: A host computer that simulates the driving behavior, the data processor that collects the monitored parameter's information and the Smart Card that stores the recorded and processed data. Serial and parallel communications assume the exchange of data between the different parts of the system. A detailed description of the system is done in this chapter discussing the hardware constituents as well as the programmed code, emphasizing on the communication protocols used and some important functions.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

6.1 General

The main objective of onboard recorders is to enhance road safety by monitoring driving habits and reconstructing accidents through the recorded data. Information has been sensed, recorded and off loaded from commercial vehicles for almost twenty years, excluding the recording of speed by tachographs. The recording devices include trip recorders, engine controls, on-board computers and wireless communication equipment. They all rely on a digital memory that remains inside the truck at all times except for the tachograph, which has proven not to be very reliable. In this dissertation, a concept of an onboard recorder for commercial vehicle application is designed and developed. Its particularity is the data storage medium that is employed, for it stores the recorded information in a digital format onto a Smart Card that is kept with the driver at all times, thus allowing drivers total confidentiality.

6.2 Major Highlights of The Investigation

The major highlights of this investigation are summarized as follows:

(1) Several bad driving habits are evoked, namely drowsiness while on the road, close driving, over speeding and lane changing. An investigation of each of the four-cited manoeuvres is undertaken in order to determine their probable causes and find solutions to prevent related road accidents. In addition, a range of frequency from 0.25 to 1 Hz is defined as representing the steering wheel motion that characterizes a non-drowsy state. A Domain

characterizing a safe lane change is also determined showing the changes with increased speed.

(2) Onboard recorders are discussed, mainly the Black Box and the tachograph. The investigation is focused on the advantages they offer to the enhancement of road safety, their multiple uses as well as their defects, showing their degree of liability versus fraud and tampering.

(3) The Smart Card monitor is proposed as a replacement and a solution to onboard recorders in the automotive industry emphasizing the benefits of working with a Smart Card and the advantages that the system offers.

(4) A thorough investigation is undergone on smart cards, discussing all its parts and explaining the several steps required for programming the chip, namely the personalization stage, the authentication required for the implemented security and the file structure and settings. The several aspects of the ACOS1 Smart Card are discussed along the main lines of smart cards noting its extra features as well as its formatting method. The several communication protocols, used to transfer data from and to the card, are investigated and discussed in detail.

(5) A monitoring system similar to the data recorders available on the market is developed for commercial vehicle use. The device uses a PIC16F877 microcontroller to process incoming information from sensors through interfacing IC chips, and stores the data onto the Smart Card. A computer-generated program simulates the speed data whereas a sensor generates the lateral accelerations that the vehicle is exposed to while driving.

(6) Communication between the monitoring system and the external sensor, the simulating computer and the smart card terminal are developed respectively using analogue to digital conversion, parallel and serial communication protocols, mainly the ISO 7816-1/2/3/4 and the RS-232 standards.

6.3 Conclusion

Based on the results of the investigation carried out in this thesis, the following conclusions are drawn:

1. The literature review has shown the impact of fatigue, drowsiness and speed on road safety and the statistics that have been found in relation to road crashes. It is also apparent that the widespread use of the onboard records and the several upgrades that have been available so far, along with the other proposed solutions and regulations, helped to enhance the security on the roads.
2. The main objective of the thesis is to design and construct a low cost system based on Smart Card, capable of monitoring speed and electronic logbook in a truck.
3. The main concerns are focused on commercial vehicles and the drivers driving habits. An investigation on drowsiness is conducted and the results obtained show that the steering wheel motion could be monitored in order to predict certain drowsy condition. A further study is conducted on the effects of speed on road safety and simulations using TruckSim software show the effect of speed on cornering, braking distance and rollover. A lane change domain is also found for

commercial vehicles in order to predict such safe manoeuvres and delimitate the acceptable trajectories that could be negotiated.

4. The proposed solution to enhance road safety utilizes the Smart Card, a card similar to the widely known credit card having an integrated chip on it. An electronic system, controlled by a PIC16F877 microcontroller monitors the speed, the lateral acceleration and the driving time of a truck, it encoded the processed data and stores it in the Smart Card.
5. The implementation of this system could help build a large database if related to speed or driving time and real conditions could lead to better road codes. Same advantages could be extended to learn more about the real driving habits of the truck drivers.
6. The system as proposed is capable of measuring and recording information related to measurable physical quantities, which are in some cases more relevant than others.
7. Since IVS target is road safety, the measurement is intended to bring as relevant as possible information. Since the lateral stability of the vehicle is related to the threshold acceleration, the detection and recording of the lateral acceleration could provide many useful information about the previous driving conditions of the vehicle.

6.4 Recommendations For Future Work

The thesis presents the results of the design of an onboard recording system used to measure driving parameters. In view of the excellent potentials demonstrated in this study, it is recommended to undertake the following future work to further explore the performance potentials and the extra features that the device can offer.

(1) The memory capacity of the Smart Card used for data recording should be increased from 8K to 16K or more in order to expand the system to be able to monitor additional parameters.

(2) Extra sensors should be added to the system in order to monitor additional parameters on the vehicle as for example a gyroscope, a longitudinal accelerometer, etc.

(3) Detail analysis on the location of the sensors onboard the vehicle requires detail analysis.

(4) The extra sensors and memory will allow the card to be used along with the data recorder, as a medical card that could help save drivers in serious accidents and a driving license card to help reduce the number of cards carried by drivers.

(5) Wireless communication using Bluetooth technology could be implemented on the system in order to allow the vehicle to communicate with its surroundings.

(6) The system should be tested using a real truck.

REFERENCES

[1] "Advocates of Highway and Auto Safety, Fact Sheet, Truck Driver Fatigue", http://www.saferoads.org/issues/fs-truckdriverfatigue.htm, 2001.

[2] Bernard, T.M., Bouck, L.H., Young, W.S., "Stress Factors Experienced by", ASSE Foundation Research Report, September 2000.

[3] Road Safety and Motor Vehicle Regulation Directorate, "Heavy Truck Collisions 1994–1998", TP 24626 E, Fact Sheet # RS 2001_05 Transport Canada, December 2001.

[4] Grace, R., Byrne, V.E., Bierman, D.M., Legrand, J., Gricourt, D., Davis, B.K., Staszewski, J.J., and Carnahan, B., "A Drowsy Driver Detection System for Heavy Vehicles," Proceedings of the 17th Digital Avionics Systems Conference, Vol. 2, 2001, pp. I36/1 - I36/8.

[5] Land Transport, The European Commission. "Road Safety", http://europa.eu.int/comm/transport/road/roadsafety/index_en.htm, updated 09/08/2002.

[6] Evans, B., "The monthly newsletter of Canadians for Responsible and safe highways", Crash Communicator, April/ May 2000.

[7] Crolla, D. A., Whitehead, J. P., "Reliable Subjective Assessment of Vehicle Handling by Drivers – Is IT an elusive goal?", Proc. ASME International Mechanical Engineering Congress and Exhibition, New York, Session DE-1B. 2001.

[8] Chan, S-C.C., "An Overview of Smart Card Security", http://home.hkstar.com/~alanchan/papers/smartCardSecurity/ , 1997

[9] "Transportation in Canada 2000 - Annual Report", Canada, Minister of transport, chapter 2. http://www.tc.gc.ca/pol/en/Report/anre2000/tc0000ae.htm

[10] "Heavy Goods Vehicles in Road Accidents: Great Britain 1998", Department for Transport, Transport Statistics, 24 October 2002.

[11] Robert, M., Gonthier, M., Mercier, P., Poulin, H., Turcotte, A., *Report of the Government-Industry Advisory Committee on Heavy Vehicle Safety: submitted to the minister transports du Québec, Monsieur Guy Chevrette*, Québec: Ministère des transports, Direction des communications, 1999. 44p.;28 cm, ISBN: 2-550-34944-X (br.): gratuity. T73A1T325A

[12] Stuster, J.W., "The Unsafe Driving Acts of Motorists in the Vicinity of Large Trucks", Anacapa Sciences INC, Contract number DTFH61-97-C-00090, February 1999.

[13] Monaco, K., Williams, E., "Assessing the Determinants of Safety in the Trucking Industry", Department of Economics, University of Wisconsin-Eau Claire, Journal of Transportation and statistics, April 2000.

[14] Evans, B., "The Monthly Newsletter of Canadians for Responsible and Safe Highways", Crash Communicator, June/July 2000.

[15] King, D.J., Mumford D.K., "An algotithm for Detecting Heavy-Truck Driver Fatigue from Steering Wheel Motion", MacInnis Engineering Associates Ltd., Canada, Paper Number 98-S4-O- 10, p873-882.

[16] Gibons J. H., "Gearing Up for Safety: Motor Carrier. Safety in a Competitive Environment", NTIS order #PB89-124796, September 1988.

[17] McDonald N., Akerstedt T., Garo C., Hamelin P., Van Ouwerkerk F., "The Role of Driver Fatigue in Commercial Road Transport Crashes", European Transport Safety Council, ISBN: 90-76024-09-X, Brussels 2001.

[18] Wylie D., Shultz T., Miller J.C., Mitler M.M., Mackie R.R., "Commercial Motor Vehicle Driver Fatigue and Alertness Study: Technical Summary", (Report No. FHWA-MC-97-001), Washington, D.C.: Federal Highway Administration, 1996. (French translation available from Transport Canada, "Etude sur la Fatigue et la Vigilance chez les Conducteurs de Vehicules Utilitaires - Sommaire technique," Report No. TP 12876E).

[19] Haworth N., Heffernan C., Horne E., "Fatigue in Truck Accidents", Monash University accident research center, Australia, ISBN 0 7326 0002 2 – Report #3, February 1989.

[20] "Driver Fatigue and Road Accidents. A Literature Review and Position Paper", The Royal Society for the prevention of accidents (ROSPA). United Kingdom, http://www.rospa.org.uk/pdfs/road/fatigue.pdf, February 2001

[21] "Managing Fatigue and Transport, Inadequate sleep and traffic accidents", A submission to the house of parliaments standing committee on communication, transport and the arts, parliament of the commonwealth of Australia, http://www.mainroads.wa.gov.au/NR/mrwa/internet/safety/driver_fatigue/others/f_transport.pdf

[22] Loss Control Department, "Fatigue: Its Impact On Motor Vehicle Accidents", The Hartford Loss Control Department, Technical Information Paper Series, TIPS S 331.002, 2000.

[23] "Motor Vehicle Transport Act, 1987. Commercial Vehicle Drivers Hours of Service Regulations", Transport Canada, http://laws.justice.gc.ca/en/M-12.01/SOR-94-716/145879.html, 1994.

[24] "Driving hours and logbooks", Fact sheet 2 April 2001, http://www.ltsa.govt.nz/factsheets/02.html

[25] Jackson, O., "Commercial Vehicle Hours of Service", Interim Report of the Standing Committee on Transport, Canada, http://www.highwaystarmagazine.com/PCJuneReport.pdf, June 2002.

[26] "Draft Regulation of the European Parliament and of the Council on the harmonisation of certain social legislation relating to road transport", Select Committee on European Scrutiny, Eighteenth Report, 12 October 2001.

[27] Circuitlink International, "*The TACHOLINK*", http://www.nlc.net.au/~circuitlink/main.htm.

[28] Menig, P., Coverdill, C., "Transportation Recorders on Commercial Vehicles", International Symposium on Transportation Recorders, Arlington, Virginia, May 3 - 5, 1999.

[29] TVI Europe Limited, "*Automatic Electronic Tachograph*", http://www.tvieurope.com/8400.html, 2003.

[30] Innovationhouse.com, "*OBD II Autowatch*", http://www.innovationhouse.com/products/autowatch_models.html, 1999.

[31] Dole, L., "On-board recorders, the Black Boxes of the Trucking Industry", International Symposium on Transportation Recorders, Arlington, Virginia, May 3 - 5, 1999.

[32] Ukawa, H., Idonuma, H., Fujimura, T., "A Study on the Autonomous Driving System of Heavy Duty Vehicle", Int. J. of Vehicle Autonomous systems (IJVAS), Vol. 1, No. 1, 2002.

[33] Grace, R., Byrne, V.E., Bierman, D.M., Legrand, J-M., Griscourt, M., Davis, R.K., Staszewski, J.J., Carnahan, B., "A Drowsy driver detection system for heavy vehicles", IEEE 1998.

[34] Jurgensen, T.M., Guthery, S.B., *Smart Cards, The developer's Toolkit*, Prentice Hall, 2000.

[35] W. Rankl & W. Effing, *Smart Card Handbook*, second edition, John Wiley & Sons, LTD, 2000.

[36] Chaum, D., "Prepaid Smart Card Techniques, A Brief Introduction and Comparison", http://ntrg.cs.tcd.ie/mepeirce/Project/Chaum/cardcom.html, 1994.

[37] Hall, E.S., Vawdrey, D.K., Knutson, C.D., "The Poket Doctor, A Bluetooth Healthcare Application", Computer Science Department, Brigham Young University, Provo USA.

[38] Nakamura, K., Hondo, I., Hataoka, N., Horii, S., "Car Information Systems for ITS", Hitachi Review Vol. 49 (2000), No. 3 Firewall.

[39] Itoi, N., Honeyman, P., "Practical Security Systems with Smart Cards", Center of Information Technology Integration, The University of Michigan Ann Arbor, 1999.

[40] Huang, M., *Vehicle Crash Mechanics,* CRC Press LLC. 2002.

[41] Brooks, C., "Speed and Heavy Vehicle Safety", Australian Transport Safety Bureau, National Heavy Vehicle Safety Seminar, Melbourne, October 2002, p.39.

[42] "Heavy Vehicle Stability Guide", Land Transport Safety Authority. New Zealand. Version 1.0. http://www.itsa.govt.nz/publications/docs/heavy-vehicle-guide.pdf.

[43] Gilman, D., "Automotive Black Box Data Recovery Systems", Ventronix Corporation, http://www-nrd.nhtsa.dot.gov/edr-site/uploads/Auto_Black_Box_Data_Recovery_Systems_by_TARO.pdf

[44] German, A., "Event Data Recorders: A New Resource for Traffic Safety Research?", Transport Canada, The Official Newsletter of the Canadian Association of Road Safety Professionals (CARSP), November 2000 Issue.

[45] "Current Trends in vehicle 'Black Box' technology", Detroit Diesel Corporation, DDEC Report. http://www.accidentresearch.com/edr-flyer.pdf.

[46] Schmidt-Cotta, R-R., "The Use of UDS-Black Box in Car Accidents, Technical, Legal, Economical and Political Aspects", PEOPIL Conference, Roma 24-25 May 2002.

[47] Anderson, R., "On the Security of Digital Tachographs", 5th European Symposium on Research in Computer Security (ESORICS), 1998, p.111-125

[48] "Smart cards for trucking industry", Giesecke and Devrient press release, http://www.gdai.com/gdai-pr/press080602.htm, August 6, 2002.

[49] Everett, D.B., "Smart Card Technology: Introduction to Smart Cards", Technical Advisor to Smart Card News, Smart Card News Ltd., Brighton, England, 1998.

[50] "Smart card technology – A tutorial", http://egov.gov/smartgov/scpm.cfm.

[51] "Study On the Feasibility and Cost of the Implementation of Intelligent Transportation Systems to Heavy Vehicles to Enhance the Road Safety". NATQ MTQ-SAAQ, Interim Report, April 1, 2003.

[52] *Smart Cards. A Guide to Building and Managing Smart Card Applications,* Henry Dreifus & J. Thomas Monk, John Wiley & Sons, 1999.

[53] *ACOS1 Smart Card with 8K EEPROM*, Reference Manual, Advanced Card Systems Ltd. 2000.

[54] ISO/IEC 7816-3 Amendment 2, Identification Cards-Integrated Circuit(s) Cards With Contacts, 1994-12-01.

[55] PIC16F87X Data Sheet 28/40 Pin 8-Bit CMOS FLASH Microcontrollers, Microchip Technology Inc. 2001.

[56] PICmicro™ Mid-Range MCU Family Reference Manual, Microchip Technology Inc. 1997.

[57] Sid Katzen, *The Quintessential PIC Microcontroller,* Springer-Verlag London Limited 2001.

[58] MPASM User's Guide with MPLINK and MPLIB, Microchip Technology Inc. 1999.

[59] D'Souza, S., "AN548 Implementing Table Read and Table Write", Microchip Technology Inc. 1997.

[60] Hai, T.B., "TB028 Technique to Calculate Day of Week", Microchip Technology Inc. 1998.

[61] Palmer, M., "AN582 Low-Power Real-Time Clock", Microchip Technology Inc. 1997.

[62] Palmer, M., "AN580 Using Timer1 in Asynchronous Clock Mode", Microchip Technology Inc. 1997.

[63] Palacherla, A., "AN547 Serial Port Utilities", Microchip Technology Inc. 1997.

[64] Fink, S., "AN579 Using the 8-bit Parallel Slave Port", Microchip Technology Inc. 1997.

[65] Pinto Souto, J.L., H & S Projetos, Florian Upolis Brazil, "Asynchronous Serial Transmit and Receive", Microchip Technology Inc. 1997.

[66] Palacherla, A., "AN510 Implementation of an Asynchronous Serial I/O", Microchip Technology Inc. 1997.

[67] MAXIM, +5V-Powered, Multi-Channel RS-232 Drivers/Receivers. Maxim Integrated Products, 1996.

[68] Fairchild Semiconductor Corporation. DM74LS245 3-State Octal Bus Transceiver, August 1986, Revised March 2000.

[69] Gadre, D.V., *Programming the Parallel Port, Interfacing the PC for Data Acquisition and Process Control*, Miller Freeman Inc. 1998

[70] Curtis, C., "Gearing Up for Safety: Motor Carrier Safety in a Competitive Environment", NTIS order #PB89-124796, September 1988.

[71] Liu, P.J., *Analysis Detection and Early Warning Control of Dynamic Rollover of Heavy Freight Vehicles,* PhD of Engineering Thesis, Concordia University, Department of Mechanical and Industrial Engineering, 1999.

APPENDIX

The appendix contains information related to the PIC16F87X family, its core and peripheral features, its block diagram and its several registers.

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

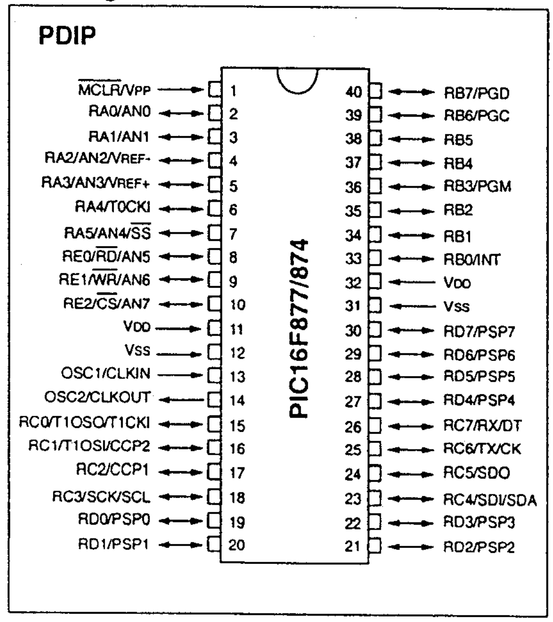
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
  Up to 368 x 8 bytes of Data Memory (RAM)
  Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 μA typical @ 3V, 32 kHz
  - < 1 μA typical standby current

Pin Diagram



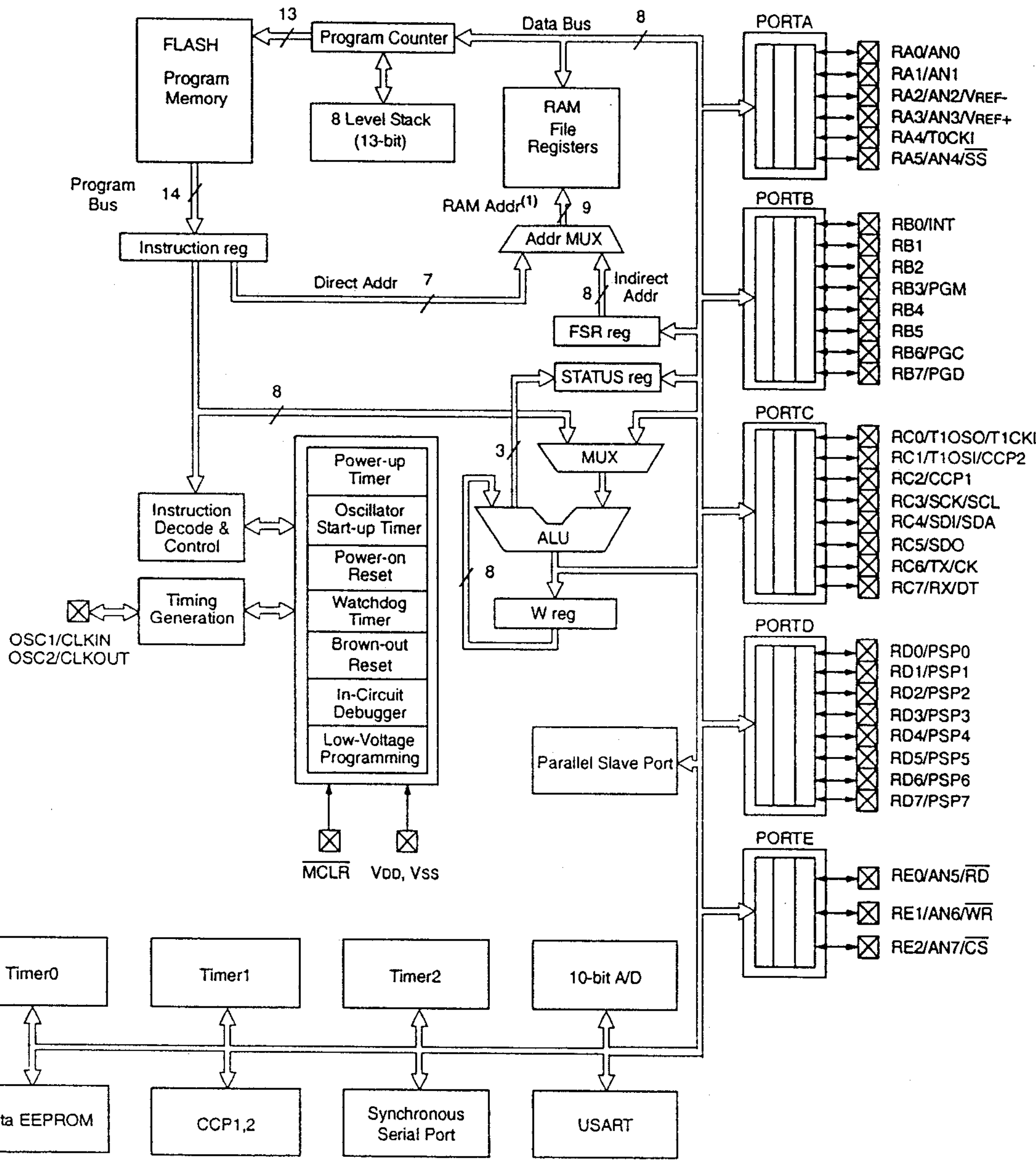
Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^{2}C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Note 1: Higher order bits are from the STATUS register.

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral features section.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
Bank 0											
00h[3]	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
01h	TMR0	Timer0 Module Register								xxxx xxxx	47
02h[3]	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
03h[3]	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	18
04h[3]	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	29
06h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	31
07h	PORTC	PORTC Data Latch when written: PORTC pins when read								xxxx xxxx	33
08h[4]	PORTD	PORTD Data Latch when written: PORTD pins when read								xxxx xxxx	35
09h[4]	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	36
0Ah[1,3]	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
0Bh[3]	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
0Ch	PIR1	PSPIF[3]	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	22
0Dh	PIR2	—	(5)	—	EEIF	BCLIF	—	—	CCP2IF	-r-0 0--0	24
0Eh	TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	52
0Fh	TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	52
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	51
11h	TMR2	Timer2 Module Register								0000 0000	55
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	55
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	70, 73
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	67
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	57
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	57
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	58
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	96
19h	TXREG	USART Transmit Data Register								0000 0000	99
1Ah	RCREG	USART Receive Data Register								0000 0000	101
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	57
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	57
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	58
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	116
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{DONE}$	—	ADON	0000 00-0	111

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

3: These registers can be addressed from any bank.

4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.

5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
Bank 1											
80h(3)	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
81h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	19
82h(3)	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
83h(3)	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	18
84h(3)	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	29
86h	TRISB	PORTB Data Direction Register								1111 1111	31
87h	TRISC	PORTC Data Direction Register								1111 1111	33
88h(4)	TRISD	PORTD Data Direction Register								1111 1111	35
89h(4)	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	37
8Ah(1,3)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
8Bh(3)	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
8Ch	PIE1	PSPIE(2)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	21
8Dh	PIE2	—	(5)	—	EEIE	BCLIE	—	—	CCP2IE	-r-0 0--0	23
8Eh	PCON	—	—	—	—	—	—	$\overline{POR}$	$\overline{BOR}$	---- --qq	25
8Fh	—	Unimplemented								—	—
90h	—	Unimplemented								—	—
91h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	68
92h	PR2	Timer2 Period Register								1111 1111	55
93h	SSPADD	Synchronous Serial Port (I^2C mode) Address Register								0000 0000	73, 74
94h	SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	66
95h	—	Unimplemented								—	—
96h	—	Unimplemented								—	—
97h	—	Unimplemented								—	—
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	95
99h	SPBRG	Baud Rate Generator Register								0000 0000	97
9Ah	—	Unimplemented								—	—
9Bh	—	Unimplemented								—	—
9Ch	—	Unimplemented								—	—
9Dh	—	Unimplemented								—	—
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	116
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000	112

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

3: These registers can be addressed from any bank.

4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.

5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:
Bank 2											
100h[(3)]	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
101h	TMR0	Timer0 Module Register								xxxx xxxx	47
102h[(3)]	PCL	Program Counter's (PC) Least Significant Byte								0000 0000	26
103h[(3)]	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	18
104h[(3)]	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
105h	—	Unimplemented								—	—
106h	PORTB	PORTB Data Latch when written: PORTB pins when read								xxxx xxxx	31
107h	—	Unimplemented								—	—
108h	—	Unimplemented								—	—
109h	—	Unimplemented								—	—
10Ah[(1,3)]	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
10Bh[(3)]	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
10Ch	EEDATA	EEPROM Data Register Low Byte								xxxx xxxx	41
10Dh	EEADR	EEPROM Address Register Low Byte								xxxx xxxx	41
10Eh	EEDATH	—	—	EEPROM Data Register High Byte						xxxx xxxx	41
10Fh	EEADRH	—	—	—	EEPROM Address Register High Byte					xxxx xxxx	41
Bank 3											
180h[(3)]	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	27
181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	19
182h[(3)]	PCL	Program Counter (PC) Least Significant Byte								0000 0000	26
183h[(3)]	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	18
184h[(3)]	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	27
185h	—	Unimplemented								—	—
186h	TRISB	PORTB Data Direction Register								1111 1111	31
187h	—	Unimplemented								—	—
188h	—	Unimplemented								—	—
189h	—	Unimplemented								—	—
18Ah[(1,3)]	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					---0 0000	26
18Bh[(3)]	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	20
18Ch	EECON1	EEPGD	—	—	—	WRERR	WREN	WR	RD	x--- x000	41, 42
18Dh	EECON2	EEPROM Control Register2 (not a physical register)								---- ----	41
18Eh	—	Reserved maintain clear								0000 0000	—
18Fh	—	Reserved maintain clear								0000 0000	—

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8> whose contents are transferred to the upper byte of the program counter.

2: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

3: These registers can be addressed from any bank.

4: PORTD, PORTE, TRISD, and TRISE are not physically implemented on PIC16F873/876 devices; read as '0'.

5: PIR2<6> and PIE2<6> are reserved on these devices; always maintain these bits clear.