

March 9, 1999

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, BC
V5A 1S6

Re: ENSC 370 Project MAPSS Design Specifications

Dear Dr. Rawicz:

The attached document, *MAPSS (Mobile Alarm Paging Security System) Design Specification*, outlines the design of our ENSC 370 project. Our goal is to design and implement an automobile feedback information center. The device alerts the car owner in the occurrence of a car alarm, the activation of the ignition system, and if the headlights have been left on. The module will deliver a unique mobile page for each such event.

This document highlights the technical design of the entire system, and the detailed specifications of the various components within the system. The designed sub components include the sensor input and decoding unit, page encoding and dialing unit, the programmable user interface unit, and the power circuit unit.

Smart Sense Innovations consists of four motivated, innovative, and talented third-year engineering students - May Huang, Shirley Wong, Caroline Dayyani, and Frederick Ghahramani. If you have any questions or concerns about this document, Please feel free to contact me by phone at (604) 941-0629 or by e-mail.

Sincerely,

Shirley Wong

Shirley Wong
President and CEO
Smart Sense Innovations

Enclosure: MAPSS Design Specification

Design Specification for a Mobile Alarm Paging Security System

Project team: Caroline Dayyani
Frederick Ghahramani
May Huang
Shirley Wong

Contact Personnel: Frederick Ghahramani
fghahram@sfu.ca

Submitted to: Dr. Andrew Rawicz – ENSC 370
Steve Whitmore – ENSC 305
School of Engineering Science
Simon Fraser University

Issued date: March 9, 1999
Revision: 1.0

Executive Summary

Your car was stolen. What do you do? CALL THE POLICE! Your car's gone!! They'll catch the crooks, like they always do on television. First they'll put out an all point's bulletin, then find your car and fingerprint it for leads to find the punks... wishful thinking.

Provincial auto-crime has increased 750% in the past decade. This staggering statistic has led to the inability of law-enforcement officials to 'chase the crooks'. In most circumstances when the exact timing of the auto-crime is unknown, the police record the theft claim and description of the car for insurance purposes. No investigation, no fingerprints, no car chases. This predicament has forced consumers to purchase anti-theft systems to better safeguard their automobile investments. However traditional car alarm systems are unreliable, annoying, persistently going off, and provide little feedback to the car owner as to the state of the car.

Smart Sense Innovations is now developing the next generation of alarm systems. This new product informs the car owner, through a mobile page, of the state of various systems in the car. The device known as MAPSS, is a micro-controller and mobile telephone based apparatus. Sensors are attached to the car security system, the car ignition system, and the car headlights. Upon activation, the module senses if the car alarm has been activated, or if the headlights have been left on, or if the car is turned on. The device then pages the car owner accordingly, informing the owner of the specific event.

In designing and delivering this product, Smart Sense Innovations is establishing new foundations for car security systems. Gone are the days when car alarms would activate at random in hopes of scaring away criminals. SSI's next generation of auto-security systems aim to act as an interactive mobile information center: constantly feeding back vital statistics about the state of the car to it's owner.

This document highlights the technical design of the MAPPS system, and the detailed specifications of the various components within the system. The designed sub components include the sensor input and decoding unit, page encoding and dialing unit, the programmable user interface unit, and the power circuit unit.

Table of Contents

EXECUTIVE SUMMARY	II
TABLE OF FIGURES	V
INTRODUCTION	1
DESIGN SYSTEM OVERVIEW	2
PROGRAMMABLE USER INTERFACE	3
1) KEYPAD INTERFACE	3
2) RESET SUB-ROUTINE	4
3) INTERRUPT SERVICE ROUTINE	4
4) THE TRANSMIT SUB ROUTINE	5
5) LCD INTERFACE	5
SENSORY INPUT AND ENCODING	7
1) HEADLIGHTS SENSING	8
1.1) <i>Signal Retrieval</i>	8
1.2) <i>Signal Processing</i>	9
2) IGNITION SENSING	10
2.1) <i>Signal Retrieval</i>	10
2.2) <i>Signal Processing</i>	11
3) CAR ALARM SENSING	12
3.1) <i>Signal Retrieval</i>	12
3.2) <i>Signal Processing</i>	12
4) IMPACT SENSOR OF THE CAR ALARM	12
4.1) <i>Signal Retrieval</i>	12
4.2) <i>Signal Processing</i>	12
5) SIGNAL ENCODING	13
5.1) <i>Signal Encoding Using a Micro-controller and a Software as an Encoder</i>	13
5.1.1) <i>Headlights</i>	13
5.2) <i>Signal Encoding Using a Priority Encoder</i>	15
5.3) <i>Signal Encoding Using a MUX and a Micro-controller</i>	15
PAGE ENCODING AND DIALING	17
1) TELEPHONE INTERFACE	17
2) TELEPHONE ALGORITHM	19
2.1) <i>Retrieving the Pager Number from the Dialer's AVR</i>	19
2.2) <i>Retrieving the Sensor code from the Dialer's AVR</i>	19
2.3) <i>Dialing the Pager Number and Sensor Code</i>	20
3) DELAY SUBROUTINE	20
4) SHORT TIME DELAY SUBROUTINE	21
CUSTOM POWER SUPPLY	21
TESTING ALGORITHM	24
1) PROGRAMMABLE USER INTERFACE	24
2) SENSOR INPUT AND ENCODING	24
3) PAGE ENCODING AND DIALING	25
4) OVERALL SYSTEM	25

REFERENCES.....26

APPENDIX27

 A - PROGRAMMABLE USER INTERFACE27

 B - PAGE ENCODING AND DIALING UNIT28

Table of Figures

Figure 1: MAPPS System Overview.....	2
Figure 2: Programmable User Interface Context Diagram.....	3
Figure 3: Keypad Setup	3
Figure 4: LCD Display Setup	6
Figure 5: Serial Transfer Format.....	6
Figure 6: Sensory Input Context Diagram	7
Figure 7: Sensory Input and Decoding Block Diagram	7
Figure 8: Voltage Divider Circuit for Generation of Low Voltage and Current.....	9
Figure 9: Comparator Circuit Used for Ignition Detection.....	11
Figure 10: Signal Encoding Using a Micro-controller	13
Figure 11: Signal Encoding Using a Micro-controller	14
Figure 12: Signal Encoding Using a Priority Encoder.....	15
Figure 13: Signal Encoding Using a Priority Encoder.....	16
Figure 14: Page Encoding and Dialling Context Diagram.....	17
Figure 15: System Setup of the AVR Dialer Chip.....	17
Figure 16: FET Interfaced with Keypad Grid.....	18
Figure 17: Custom Power Supply	21
Figure 18: Circuit Protection Schematics	22
Figure 19: Circuit Protection Schematics	23

Introduction

Car owners have several options in attempting to counter the recent rise in auto theft and vandalism. Many third party after market systems have been developed and marketed. These products range from physical preventative devices that lock the steering wheel, to alarm driven deterrent systems that sound a siren alerting passersby of illegal activity. However traditional anti-theft alarm based systems are unreliable, annoying, persistently going off and provide little feedback to the car owner as to the state of the car. MAPSS aims to address this shortcoming in current auto-security systems by incorporating a telephony device into the system. The inclusion of a communication framework between the car and its owner serves to best inform the owner of the current conditions of the car.

MAPSS is a standalone programmable module that monitors the state of the car security system, the ignition system, and the car's headlights. Upon sensing input from the various car subsystems, MAPSS encodes and delivers a unique mobile page to the car owner. This page is displayed in an alphanumeric representation, thus informing the car owner in plain English of the state of the car.

The purpose of this document is to describe the designed components of the MAPSS device, and to resolve Smart Sense Innovations' April 1999 deliverables. The intended audience for this document is Dr. Andrew Rawicz, Mr. Steve Whitmore, the design engineers of Smart Sense Innovations, and various external third party design consultants.

Design System Overview

The design overview of the MAPPS system is shown in Figure 1. 4 small 20-pin micro-controllers are utilized to perform the tasks desired for the system. Several small sub-circuits have been designed for the various components. Each component is described with further detail in the subsequent sections.

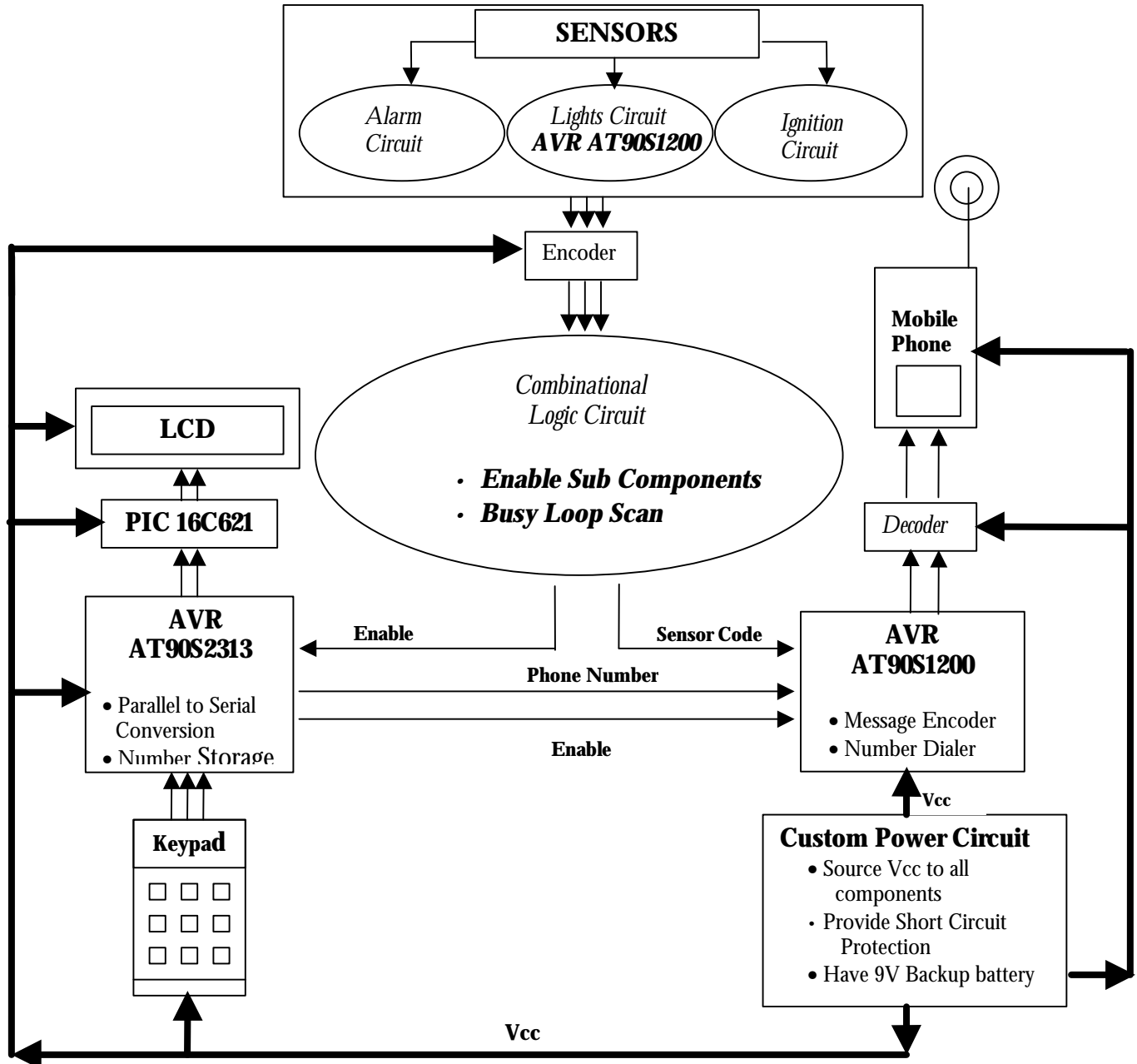


Figure 1: MAPPS System Overview

Programmable User Interface

The first stage of the MAPSS system is responsible for interfacing with the user. In specific, this stage is responsible for obtaining and storing the telephone number, which will be called at the output stage. The unit is composed of a 7-line LCD module controlled by a PIC 16C621 micro-controller, and a 12-button numeric keypad controlled by an AVR AT90S2313 micro-controller. Figure 2 demonstrates the context with which this stage fits into the entire MAPSS system.

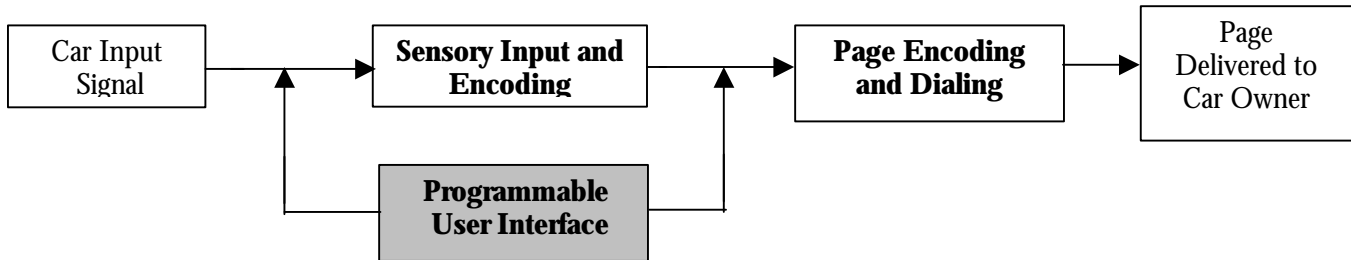


Figure 2: Programmable User Interface Context Diagram

1) Keypad Interface

The AT90S2313 used for the keypad interface spends the majority of its time in “power down mode”. This mode is instanced by the “sleep” operand sets the chip circuitry to operate in a low power consumption fashion. The watchdog timer and analog comparator are disabled. The AVR wakes up from sleep mode when a key is pressed and begins the asynchronous data transmission to the PIC of the LCD module. Figure 3 shows the keypad setup.

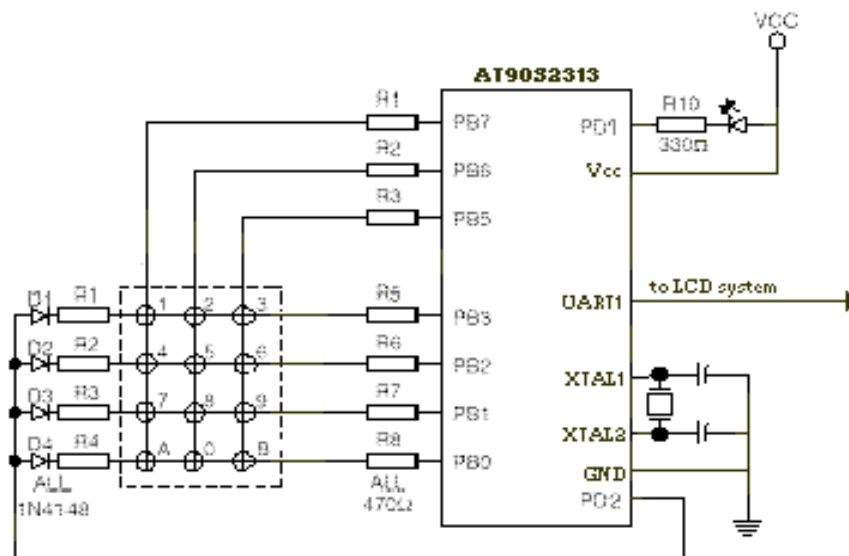


Figure 3: Keypad Setup

The keypad columns are connected to the high nibble of port B. The keypad rows are connected to the low nibble. Resistors R1 to R8 limit the input current to a safe level in the event of a short in the keypad. This may never occur, and the resistors might be removed for the final product time permitting. In “sleep mode” the high nibble is configured as output and is in the low state. The low nibble is configured as input and has the internal pull-ups enabled. After initialization the AVR is put to sleep. When a key is pressed, one of the diodes D1-D4 pulls down the external interrupt line PD2, which also has its internal pull-up enabled. This wakes up the AVR and forces it to run the interrupt service routine, which scans the keypad and calculates which key is pressed. The AVR then returns to the main program and transmits the key pressed to the LCD module through the UART port in a serial fashion. Resistor R10 (330ohm on a 5 V supply) is the current limit resistor for the LED.

The firmware consists of three sections, the reset routine, transmit routine and the interrupt service routine which sets up the ports, sleep mode, power saving and the interrupts. The assembler has been written in a modular manner. This allows us to reuse the same interrupt driven code by the other micro-controllers in the MAPPS system.

2) Reset Sub-Routine

In reset the bi-directional ports are initialized with their starting directions. These are fixed on port D, with all bits as outputs except PD2, which must be input for the external interrupt. This bit has its pull-up enabled by setting bit 2 of Port D. The unused bits are configured as outputs to avoid noise pickup.

Port B starts with the high nibble set as outputs sending out zeroes, and the low nibble set as inputs with the pull-ups enabled. This is achieved by configuring the Data Direction Register with “1”s for outputs, “0”s for inputs, and then writing “1”s to the input bits in the PORT register. The inputs can then be read or tested from the PIN register. The Transmit subprogram scans for “0”s and uses the SBIS instruction to skip over the key-press action if a “0” is not found.

Power down mode is selected by setting the SE and SM bits of the AVR. At the same time writing “0”s into the ISC00/01 bits configures the external interrupt. This will set the external interrupt INT0 to trigger on a LOW level. When using “power down” mode the AVR can only be woken up by LOW level trigger. Transmit disables global interrupts until the program is ready to be interrupted. The AVR then enters sleep mode. This action is placed in the main program loop to ensure that the circuit goes back to sleep after it has finished its interrupt function and finished the data transmission. When the AVR wakes up after a key-press, the Transmit routine is called. When the Transmit routine is done, the external interrupt is enabled, so another interrupt can occur.

3) Interrupt Service Routine

The interrupt service routine executes upon the occurrence of an external interrupt. In the case of the keypad, an interrupt occurs when PD2 is pulled low. Upon entry into the routine, the status register is preserved to avoid corrupting any work the main program was doing. The flowchart is shown in Appendix A under “Flowchart for Interrupt Service Routine”. The key row is first detected

by testing each row input, in turn looking for a “0”. A base number 0, 3, or 6 is then assigned to the variable “key”. The ports are then reinitialized with port B I/O swapped over so that the key columns can be tested. The key column is then detected and a number assigned in a temporary variable “temp” of 1, 2 or 3. Adding “key” and “temp” then calculates the final value of the key that was pressed, the result is placed in “key” ready for use by the Transmit subroutine. At the end, the external interrupt is disabled. This is done to avoid the interrupt routine being triggered again immediately upon exit. Luckily though, no human can press the keyboard so fast as to cause this to happen.

4) The Transmit Sub Routine

The flow chart for the Transmit routine is shown in the Appendices. The value of the key pressed is taken from the “key” variable and used as a pointer to access a 16-byte look-up table stored in EEPROM. The look-up table contains the number of the key pressed. The table makes the program much shorter, and allows easy extension to provide full ASCII coding for the key value. The key value derived from the EEPROM is used as a countdown variable for arranging and queuing the ASCII character (in this case a number) to the UART register. Data transmission is initiated by writing the data to be transmitted to the UART I/O Data Register (UDR). Data is transferred from the UDR, to a transmitting shift register, when a new character has been written to the UDR after the stop bit from the previous character has been shifted out. The output is fed directly into the UART port of the PIC 16C621. For the transmission between the two chips, both chips need to be transmitting and receiving at the same baud rate. The AVR AT90S2313 has an on-chip baud rate generator. The generator is a frequency divider, which generates a baud rate according to equation (1).

$$\text{Baud Rate} = \frac{f_{ck}}{16(UBRR+1)} \quad (\text{Eq. 1})$$

Where f_{ck} is the give crystal clock frequency, and UBRR represents the contents of the UART Baud Rate register (0-255). The baud rate for the transmission to the LCD module will be fixed at 1200 at a 2MHz clock, yielding a 0.2% error margin.

5) LCD Interface

The LCD interface is composed of a PIC 16C621 micro-controller that interfaces with a Hitachi HD44780 LCD module. As show in Figure 4 below, the data that is to be displayed on the LCD module is input into the PIC micro-controller in a serial fashion from the AVR controller of the keypad module.

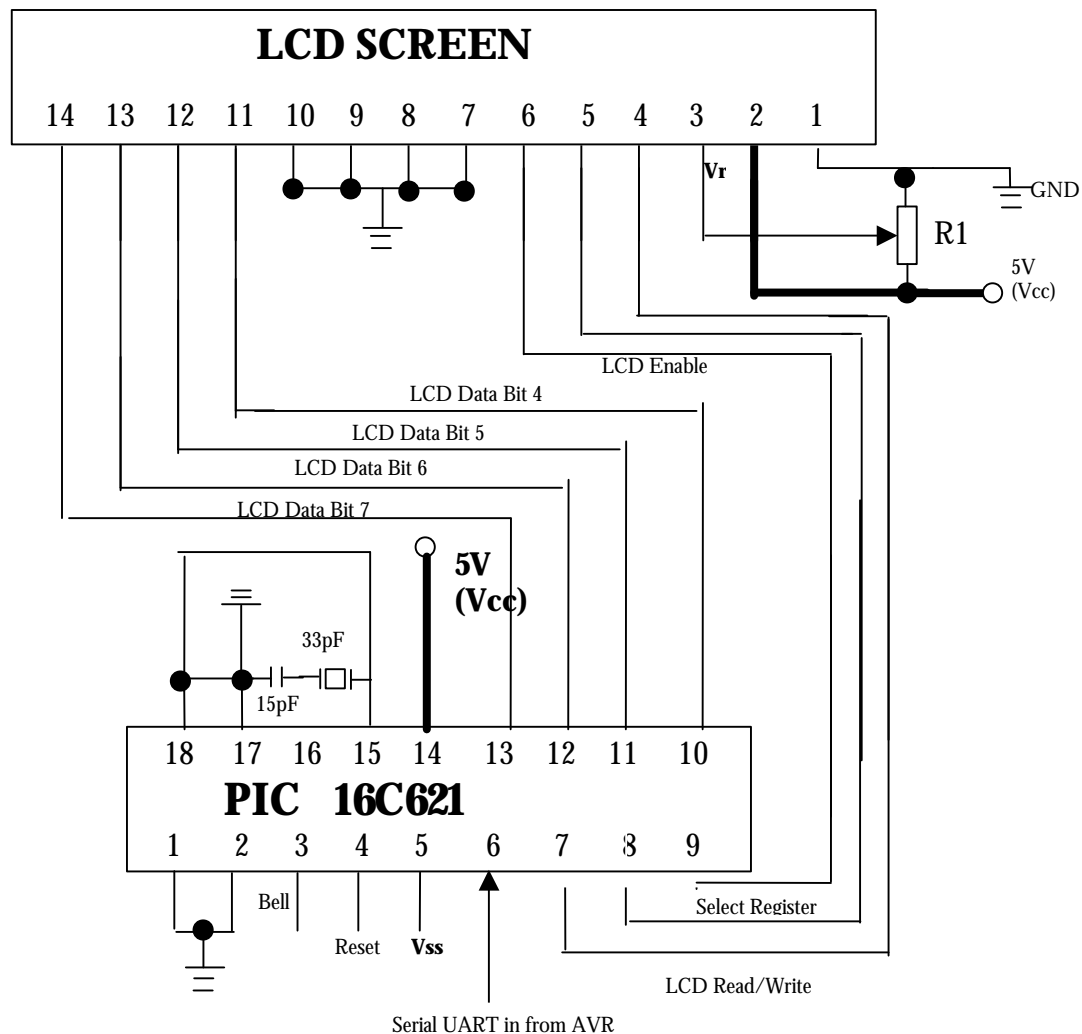


Figure 4: LCD Display Setup

The serial data is encoded using a common baud rate, and an 8-N-1 protocol. 1 start bit, 8 data bits, and 1 stop bit. A logic-timing diagram of this output is shown in Figure 5.



Figure 5: Serial Transfer Format

Sensory Input and Encoding

Figure 6 demonstrates the context with which this stage fits into the entire MAPSS system.

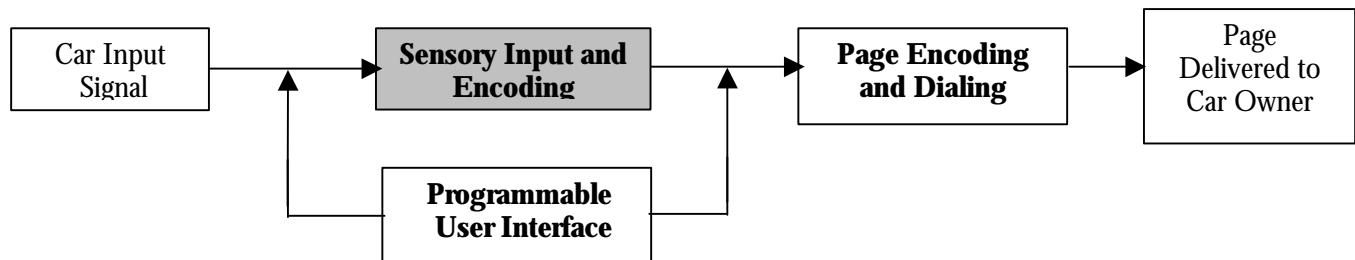


Figure 6: Sensory Input Context Diagram

Figure 7 shows more details of the Sensory Input and Decoding part of the MAPSS. Each different sub sections will be discussed in more details in the following sections.

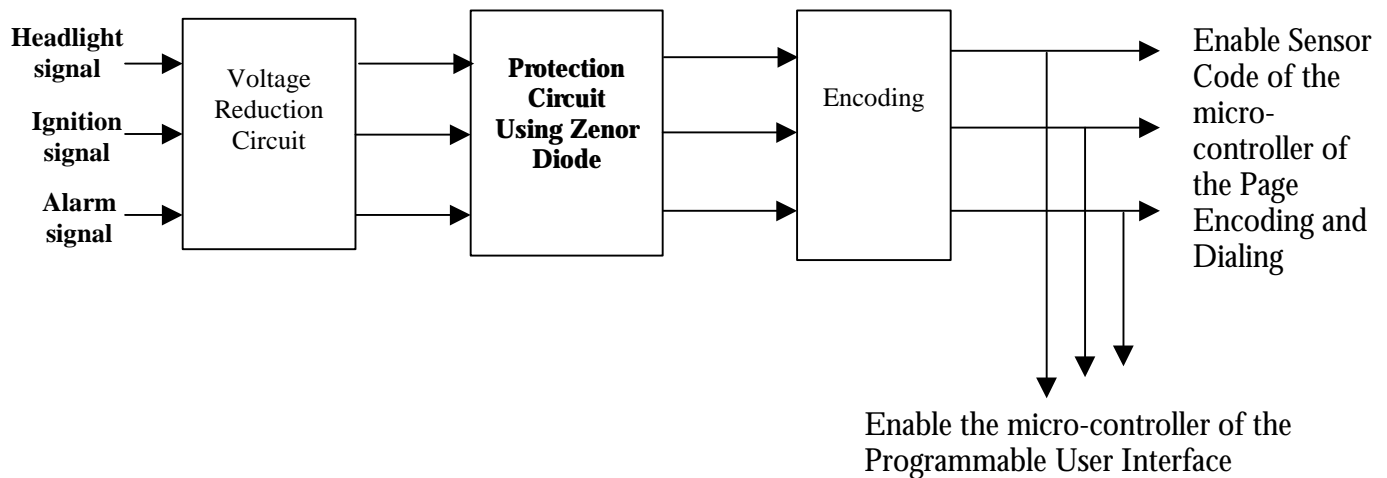


Figure 7: Sensory Input and Decoding Block Diagram

Once the MAPSS is enabled by the user, the signals from headlights, ignition, and alarm will be periodically checked, and it will be sent to the voltage reduction circuit in order to lower the voltage and the current down to a suitable value for the input of the micro-controller. In each part there is a protection circuit designed to avoid any overloading of the current or the voltage. Then the encoding part will make the encoded signals ready for the next parts of the MAPSS system.

The output of the Sensing Input not only enables the Programmable User Interface, but also it sends sensing codes to the Page Encoding and Dialing Unit so that the user will know which of the three systems are in the case of danger.

The timing circuit specific to the headlight sensing part and the voltage and current reduction circuits for each of the three parts of the Sensing Input and Decoding is explained in more details in the following sections.

1) Headlights Sensing

1.1) Signal Retrieval

The purpose of this part is to inform the owner of the car if he accidentally left the headlights on for more than 14 minutes. This part of the MAPSS reduces the consumption of the car battery to a minimum.

There are several possibilities to retrieve the signal from the headlights of the car. However, there are pros and cons in each method. The following are different possibilities, which have been considered for retrieving the headlight signal:

1. Getting signal from the built-in car micro-controller

This method of retrieving the signal from the headlights is the most clean and easy way. It requires no messing with the wires inside the car and the signal has a low voltage and current characteristics that it does not require anymore electronics circuit for the additional reductions in these parameters. However, there are some consequences in getting the signal from the micro-controller, which make this method not feasible for this project. Depending on the model of the car, the micro-controllers are different, so installing the MAPSS for different cars requires the knowledge of the internal circuitry of the micro-controllers. As well if the MAPSS system is to be marketed in the future, it would have to be installed in the car as a customized product while the car is being built. Thus MAPSS system would not remain a third party product.

2. Getting signal from the fuse box under the hood of the vehicle

This method allows us to bypass car manufacturers and same as the first method, it is an easy and clean way of receiving the signal. However, connecting anything to the fuse box is an unsafe task. The fuse box is connected to some other components in the circuit including relays, so connecting a circuit in between may alter the operation of the relay circuit and as a result the other parts of the car could be rendered inoperable.

3. Tapping the signal straight from the headlight

This method is the most feasible. It can be easily done since tapping signals are a common practice. Since the headlight is connected directly to a 12V-power supply, there is no worry of affecting any relay circuitry. But there is a need for an electronics circuit in order to reduce the voltage and the current to a desired output value of about 5V to 7V. As well tapping the signal straight from the wires under the hood, produces some distortions in the final output since the wires have to travel a long distance from the hood to the MAPSS box under the passenger seat. Even though this method still have some consequences, it is the desired method for this project considering all the available resources and accessibility to the user car.

1.2) Signal Processing

Since headlights are connected directly to a 12V battery supply and they have a consumption of about 60W nominally, a current of 5.0A goes through them. Since the output of the Sensing Input goes to a micro-controller, which requires current in the order of 10mA and a voltage of about 5V, the headlight output has to be reduced by a large factor. In order to reduce both the current and the voltage, a voltage divider circuit is used. Figure 8 shows the corresponding voltage divider circuit which his required for this part.

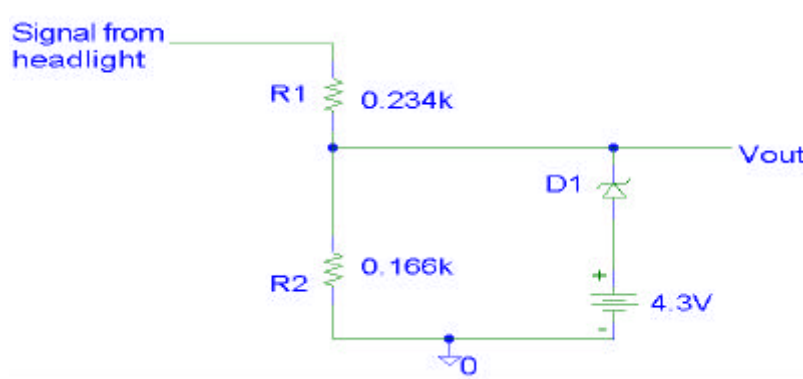


Figure 8: Voltage Divider Circuit for Generation of Low Voltage and Current

Large values of R1 and R2 are chosen somehow to avoid a large current drain into the voltage divider output and to produce a large voltage drop from the battery input to 5V. In addition, a zenor diode is placed across V_{out} for protection features. Assuming there is a 0.7V drop across the ideal zenor diode, any voltage higher than 5V will be clipped by the diode, thus this ensures the micro-controller connected at the output operates safely.

2) Ignition Sensing

2.1) Signal Retrieval

The purpose of this part is to inform the owner of the car if anyone is trying to start the car. By using the MAPSS paging system, the risk of stealing the car is minimized.

There are several possibilities to retrieve the signal from the ignition of the car. However, there are pros and cons in each way. The following are different possibilities, which have been considered for retrieving the ignition signal:

1. Retrieval of the signal by the use of a magnetic sensor

In this method the magnetic sensor is fastened on top of the motor ignition coil. Once the car starts, the large amount of current goes through the magnetic coil, which will produce a large magnetic field inside the coil. Then the magnetic sensor is able to sense this magnetic field and send the output to the enable bit of the Programmable User Interface as well as to the Page Encoding and Dialing micro-controllers. Since a sensor is used in this method, the output is already ready to be fed into the micro-controller without any further voltage regulation. As well the high sensitivity with a large bandwidth for a faster response are the other advantages of using the magnetic sensors. However, in this method, a very accurate method of fastening the sensor to the ignition coil has to be considered. In terms of the accuracy of the results, even if the sensor itself has a good signal detection percentage, the distortions coming from different parts in the neighborhood of the sensor will cause it to have less accurate results.

2. Getting signal from the built-in car micro-controller

This method of retrieving the signal is the same as the case of the headlights. It doesn't require implementation with the wires inside the car and the signal has a low voltage and current characteristics that it does not require anymore electronics circuit for the additional reductions in these parameters. However, there are similar consequences in using this method as in the case of headlights, which does not make this method a feasible way to use for this project.

3. Getting signal directly from the ignition coil

Although this method can give the required signal, reaching for the ignition coil is not an easy task to do. Since the ignition coil is inside the motor compartment, access to this coil with the limitations of this project is very hard. This method is very intrusive and is not feasible for this project.

4. Getting signal by detecting a change in the voltage of the car battery

This method is the most accurate and easy way to detect the signal of the ignition. Once the car is started, there is a drop of about 3V in the car battery from 12V to about 9V. This drop occurs in a very short period of time, resulting in a sharp spike response. The use of a comparator helps in determining the resulting change in voltage. As a result the signal will be sent to the

Programmable User Interface enable bit as well as to the Page Encoding and Dialing Unit to inform the user about the starting of the car.

2.2) Signal Processing

Since method 4 is preferred in the signal retrieval for the ignition, Figure 9 demonstrates the comparator circuit that is required for this part.

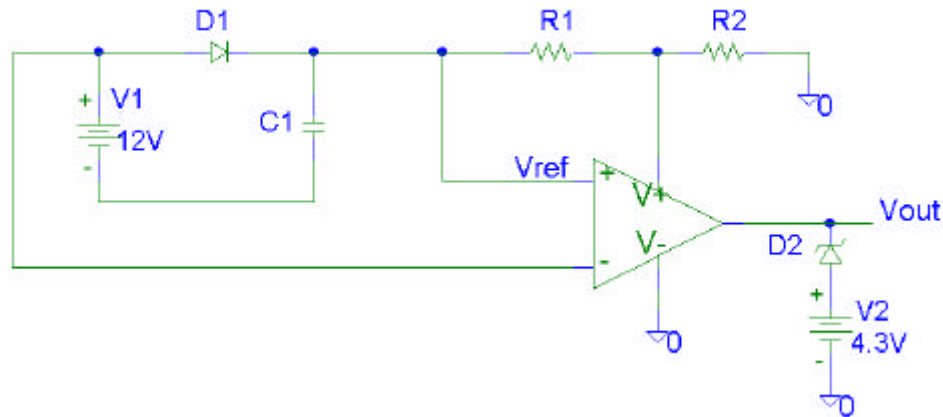


Figure 9: Comparator Circuit Used for Ignition Detection

The positive input of the comparator is called the reference voltage and it is connected to the battery of the car through a capacitor. The negative of the comparator is connected directly to the battery of the car. So whenever, the car is ignited, a drop of 3V in the car battery will be compared to the reference voltage which as a result of large capacitor time constant will not be changed very fast. Since an output of the comparator has to be 5V, then a voltage divider is used to reduce the voltage to 5V. By simple voltage divider calculations it is obvious that the ratio of R_1 to R_2 has to be 7:5. In this way the output stays at 5V and in order to protect it from any increase in the voltage, another zener diode is used.

For the comparator a LT1011/LT1011A voltage comparator is used with the maximum ratings shown in Table 1.

Table 1: Maximum Ratings of the Voltage Comparator

Supply Voltage	36 V
Output to Negative Supply	40 V
Ground to Positive Supply	30 V
Differential Input Voltage	36 V
Input Voltage (V)	Equal to Supply
Lead Temperature (soldering 10 seconds)	300°C

3) Car Alarm Sensing

3.1) Signal Retrieval

The purpose of this part is to inform the owner of the car when his car alarm system goes off so that the required action can be taken as fast as possible. In this manner using the MAPSS paging system lowers the risk of the auto crime.

In order to detect the alarm signal in the first place, the take off signal is chosen at the place where the signal goes to the siren of the alarm. The reason is that in order for the siren to start, there must be a voltage of 12V provided to it. This voltage comes from the car battery directly. As well the required current is very low (about 1A to 2A since the required power is not more than 20W) otherwise the car battery would be drained very fast. However, the voltage has to be always 12V in order for the siren to work.

3.2) Signal Processing

Since the required voltage is higher than what is desired for the output to the micro-controllers, a voltage divider similar to the one used for the headlight sensing part is required for this part as well. As it was shown in Figure 8, a zenor diode is used for the protection purposes to avoid any large values go to the output. Also large values of the resistors avoid the extra leakage of the current to the circuit, so the output current will be lower in terms of both current and voltage. However, different values for R_1 and R_2 are used in this case since the current through the siren is about 1A to 2A, then suitable values are 0.140K Ω and 0.100 K Ω respectively.

4) Impact Sensor of the Car Alarm

4.1) Signal Retrieval

The purpose of this part is to inform the owner of the car about any impacts to his car by using the impact sensor of the car alarm. Since almost all car alarms come with an impact sensor, retrieval of the signal from the sensor is similar to the case of the car alarm retrieval signal.

In order to detect the impact signal in the first place, the take off signal is chosen at the place where the signal goes to the impact sensor of the car alarm.

4.2) Signal Processing

Since the required voltage is higher than what is desired for the output to the micro-controllers, a voltage divider similar to the one used for the headlight sensing part is required for this part as well. As it was shown in Figure 8, a zenor diode is used for the protection purposes to avoid any large values go to the output. Also large values of the resistors avoid the extra leakage of the current to the

circuit, so the output current will be lower in terms of both current and voltage. However, different values for R_1 and R_2 are used in this case since the current through the siren is about 1A to 2A, then suitable values are 0.140K Ω and 0.100 K Ω respectively.

5) Signal Encoding

In order to encode the signals of the headlights, ignition, and alarm and send them to the Programmable User Interface Unit as well as the Page Encoding and Dialing Unit, there are few options available with pros and cons.

5.1) Signal Encoding Using a Micro-controller and a Software as an Encoder

Although this method requires more development time, the accuracy of the results is high enough, that there won't be any problems in terms of getting incorrect results.

5.1.1) Headlights

Figure 10 is the flow chart of the signal encoding using a micro-controller and software as an Encoder.

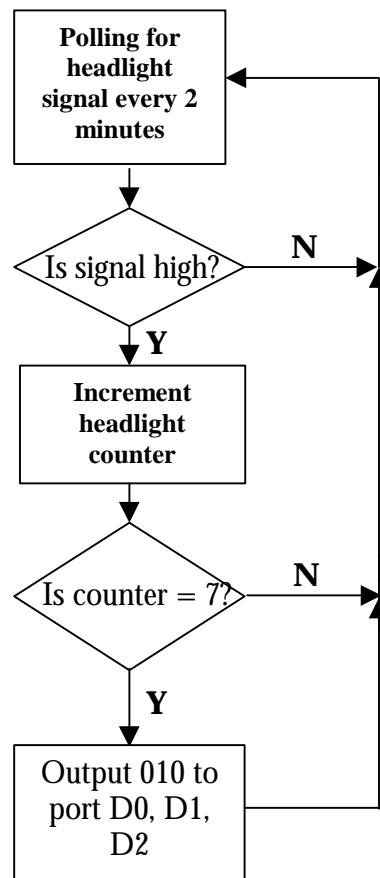


Figure 10: Signal Encoding Using a Micro-controller

The reason that this method is more accurate is that, the micro-controller periodically checks for the headlight signals every 2 minutes. Then a counter will be set so that each time the headlights were on in this 2 minutes checks, the counter will be updated so that after 7 repetition or after 14 minutes an output will be sent to the Programmable User Interface and the Page Encoding and Dialing Unit. The output is in the form of binary numbers of 010 for the headlights. The input is sent to port B of the AVR chip and the output is sent to ports D of the chip in the Page Encoding and Dialing Unit.

Finally this output will be the sensing code output for the recognition between the car alarm, headlights, and the ignition of the car. As well this output signal enables the micro-controller of the Programmable User Interface; therefor, the phone number will be transmitted to the Page Encoding and Dialing Unit.

S.4.1.2 Ignition, alarms, and impacts sensor

The encoding of the rest of the parts of the Sensory Input parts are different from the headlights since the polling method occurs more frequently since it is in a more priority order. Figure 11 shows the flow chart of this process.

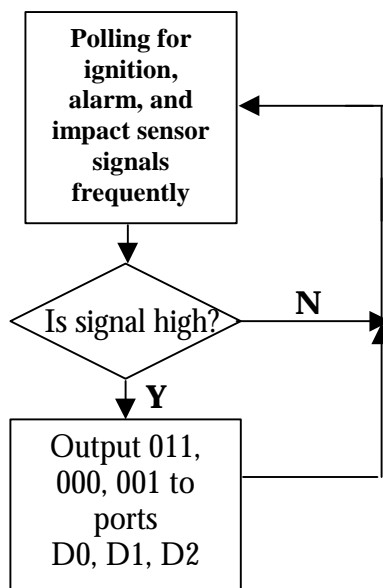


Figure 11: Signal Encoding Using a Micro-controller

Similar to the headlight polling method, the output is in the form of binary numbers of 011, 000, and 001 for the ignition, the alarm, and the impact sensor. Again the input is sent to port B of the AVR chip and the output is sent to ports D of the chip in the Page Encoding and Dialing Unit.

Finally this output will be the sensing code output for the recognition between the car alarm, headlights, and the ignition of the car. As well this output signal enables the micro-controller of the

Programmable User Interface; therefore, the phone number will be transmitted to the Page Encoding and Dialing Unit.

5.2) Signal Encoding Using a Priority Encoder

In this method, the actual signals from the headlights, ignition, car alarm, and the impact sensor will be ANDed with the priority signal from the AVR micro-controller, so that the result would only be on when both signals are on. This method eliminates the problem of the multi-signals happening at the same time since only one signal is given the priority at one time. So the accuracy of receiving the signal is higher. Figure 12 shows the signal encoding method using a priority encoder for all four signals.

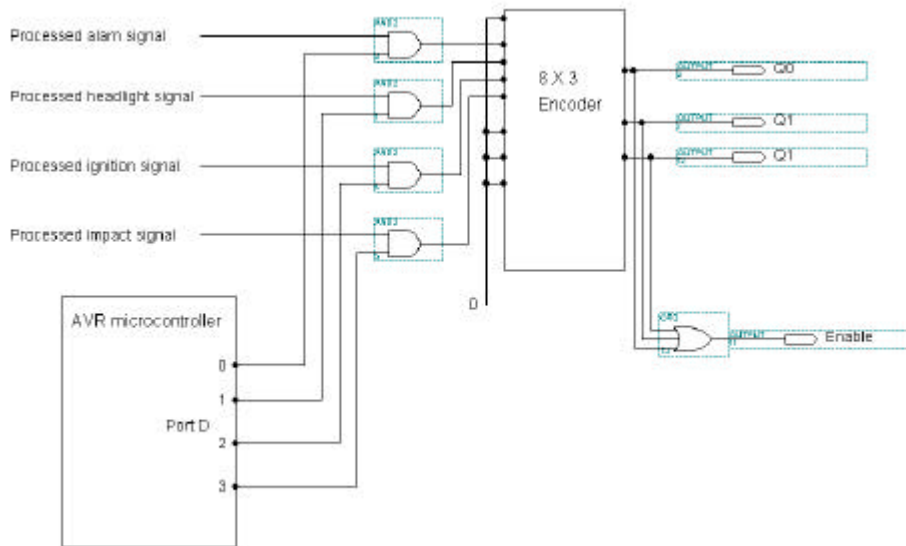


Figure 12: Signal Encoding Using a Priority Encoder

Output of the AVR from port D, is the priority signal to the AND gates and a 3 by 8 encoder is used to enable the AVR of the Programmable User Interface as well as to send the sensing codes to the Paging and Encoding unit.

5.3) Signal Encoding Using a MUX and a Micro-controller

Although this method still uses the micro-controller, it is only used to select the signal into the multiplexer. In this way the problem of the normal use of the encoder is prevented and in addition, this method requires fewer pins at the output. The output of the AVR is port D. Figure 13 shows the signal encoding using a MUX and a micro-controller.

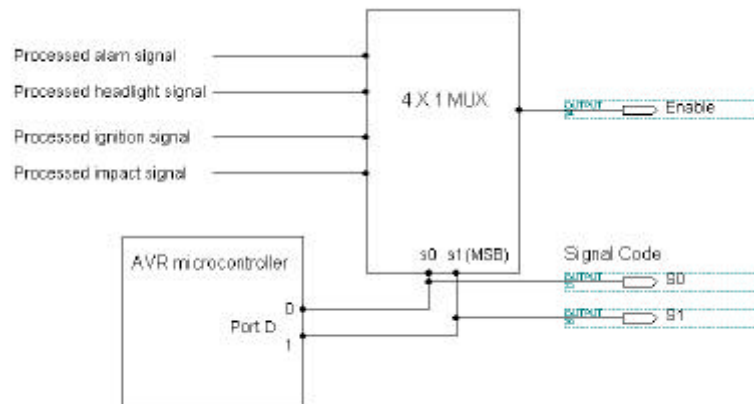


Figure 13: Signal Encoding Using a Priority Encoder

Simply the status bits of the MUX are the inputs to the sensing code of the Paging Encoding and Dialing Unit of the MAPSS. 000 for the car alarm, 001 for the impact, 010 for the headlights, and 011 for the ignition. This method is considered to be the best among all of the above methods.

Page Encoding and Dialing

The third stage of the MAPSS system is responsible for interfacing with the telephony device. In specific, this stage is responsible for encoding, composing, and delivering the appropriate mobile page. Figure 14 demonstrates the context with which this stage fits into the entire MAPSS system.

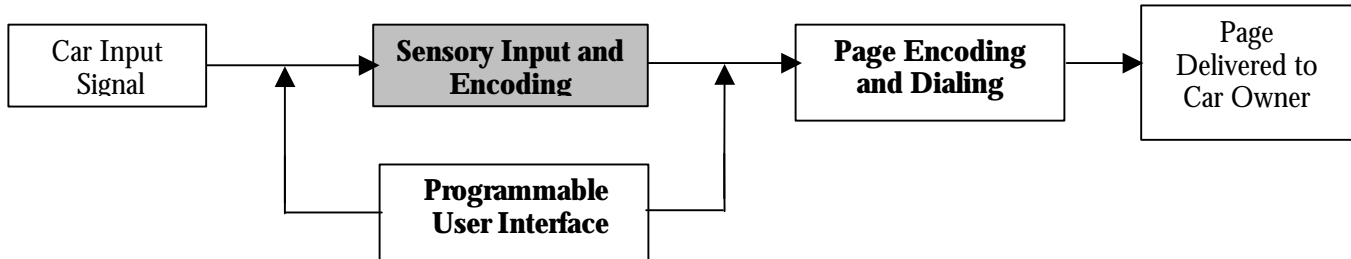


Figure 14: Page Encoding and Dialing Context Diagram

1) Telephone Interface

The MAPSS system incorporates an embedded Motorola 650 flip phone. For the purposes of the MAPSS system, the telephone will be treated as one black box PCB, with interfaced leads into the keypad. Each button of the telephone (16) is connected through a 1 of 16 decoder and a 4-line bus into the AVR ATS901200 dialer chip. The AVR chip operates in the same interrupt driven fashion as was described in the keypad section. The system setup is shown in Figure 15. The telephone keypad is interfaced through the use of a 4-line bus switch relay system.

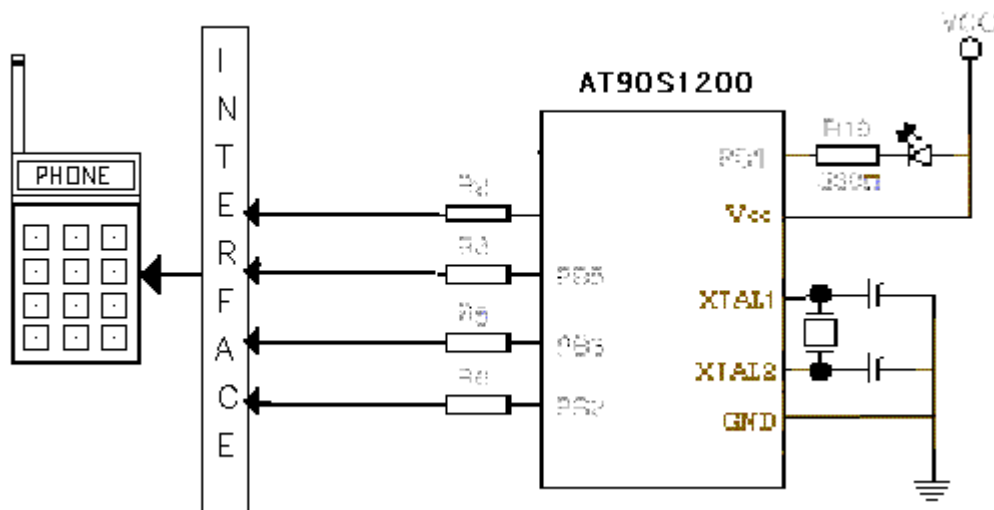
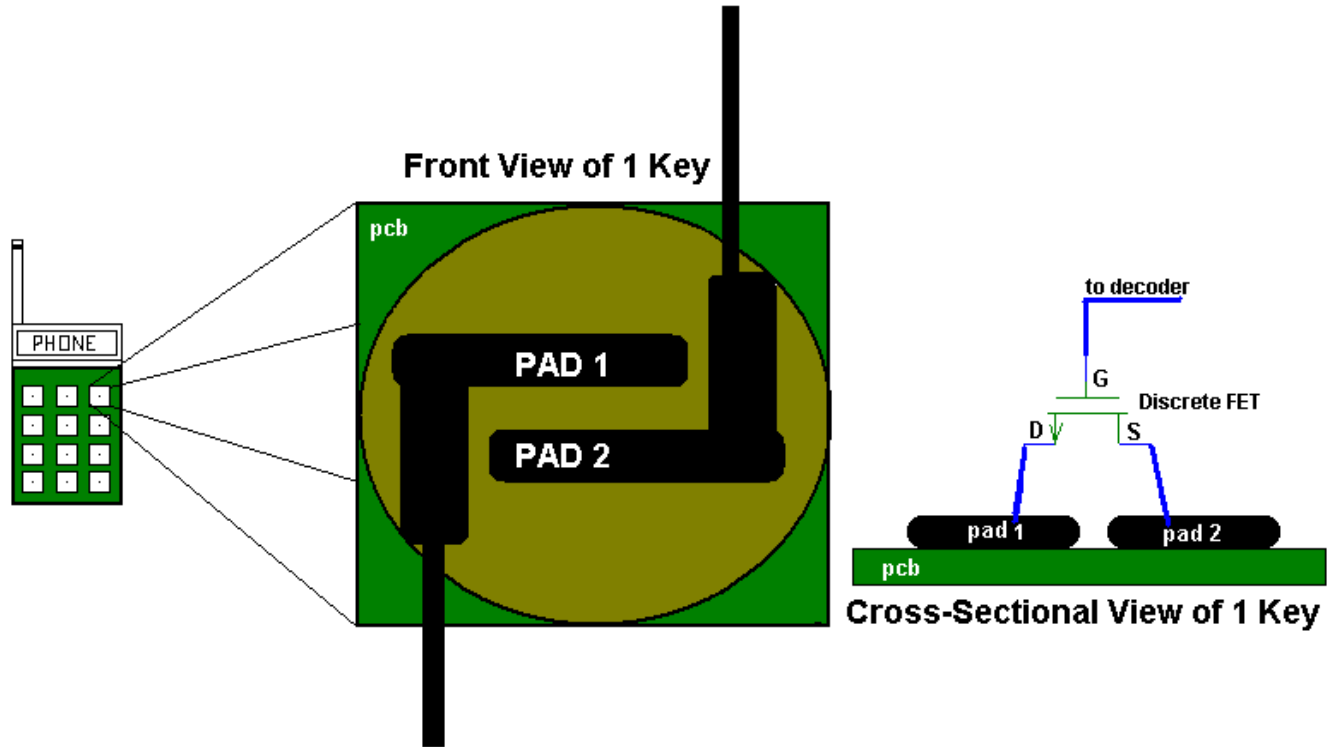


Figure 15: System Setup of the AVR Dialer Chip

At the publication of this document, the details of the interface mechanism have not been finalized. Currently the 2 possible solutions that are being considered are 1) Utilize the speaker phone input bus of the mobile phone to dial and control the phone. 2) Interface an array of surface mount



discrete FET Transistors acting as switches. The interface for option 2 is shown in Figure 16.

Figure 16: FET Interfaced with Keypad Grid

The operation of the second option is dictated by the operation of the Discrete Field Effect Transistor. The operation of the channel in the FET is dictated by Equation (2).

$$V_{DSsat} = v_{GS} - V_T \quad (\text{Eq. 2})$$

In normal modes of operation, when a button is pressed on the keypad, pad1 and pad2 of the individual button are momentarily shorted. Setting V_{gs} to the appropriate value (logic high) simulates this action. When V_{gs} drops below the cut-off point, the channel between the drain and source

becomes “pinched off”. Resulting in a potential difference between pad1 and pad2. The cut-off point of the transistor (logical low) simulates a non key-pressed situation.

2) Telephone Algorithm

2.1) Retrieving the Pager Number from the Dialer’s AVR

There are three subparts in the user interface unit of MAPSS. The first subsystem retrieves the number that is to be paged from the keypad AVR. The AVR stores the seven-digit pager number into its EEPROM. The location “\$Phone” is designated to store the number. The physical size of this memory will be 7 bytes (one byte storing one digit of the phone number). A counter is initialized as soon as the system receives an “enable” from the keypad AVR. There will be a loop that will do the storing of the pager number. The pager number is stored one digit at a time, in terms of eight bits representing that digit. A single digit will be represented as an eight-bit number. For example, the number “2” will be received and stored as “00000010”. At the end of each byte stored, the counter is incremented by one. The cycle continues until the counter equal 6, in other words, until all 7 digits have been stored. The counter ensures all eight bits of each digit are received and stored. The counter also ensures that all 7 digits of the pager number are stored inside the dialer AVR’s EEPROM. The flow chart for this subsystem can be found in the Appendix B under “Flowchart for Receiving the Pager Number from the Keypad AVR”.

2.2) Retrieving the Sensor code from the Dialer’s AVR

The user interface unit’s second task is to obtain the sensor code from the combinational logic unit. There are four different sensor codes recognized by this sub system. The four different alarms are shown in Table 2.

Table 2: Sensor Codes

Sensor Code	Sensor Type
00	Car Alarm
01	Impact
10	Headlights
11	Engine

As soon as an interrupt occurs within the dialer AVR, the micro-controller scans for the sensor code to perform the necessary algorithm for the alarm detected. The sensor code is placed by on a two-bit bus by the combinational logic unit. The dialer AVR checks the values of the bus in order to detect which sensor has been activated. After the AVR has deciphered the sensor code, it assigns values to the two variables “Code Offset” and “Code Repeat”. The value of Code Offset corresponds to the number of bytes away from the start of the sensor code storing location in the AVR’s EEPROM. We label the starting location of the sensor code storage as “\$Sensor Code”. The physical size of this location in memory is 4 bytes and it will store the number(s) to be pressed to send the alphanumeric

page for the sensor detected. (\$Sensor Code stores the binary representation of the number “2” and \$(Sensor Code + 1) stores the binary representation of the number “5”). The Code Repeat value corresponds to the number of times the number stored in memory is pressed. For example, the letter “I” requires the user to press “555”. Therefore, the “Code Repeat” value is 3. The assignments for each of the sensor codes are shown in Table 3.

Table 3: Assignments for each Sensor Code

Sensor Code	Code Offset	Code Repeat	Output on Pager
00 = Car Alarm	0	1	A
01 = Impact	1	3	I
10 = Headlights	1	2	H
11 = Engine	1	1	G

The flow chart for this subsystem can be found in the Appendix B under “Flowchart for Receiving the Sensor Code from the Keypad AVR”.

2.3) Dialing the Pager Number and Sensor Code

The last part of the User Interface Unit is the actual dialing of the pager number and alarming the user of the type of sensor that has been set off. We first define the word “delay” to be .5 seconds long and the word “long delay” to be 1 second long. After the dialer AVR has received and stored the pager number and sensor code, it will proceed onto dialing. The first step in this algorithm is to point to the location in EEPROM where the pager number is stored (\$Phone). It will decode the number (convert from the binary representation of a digit to an actual number on the dialer keypad), and then press the corresponding number on the dialer keypad. It will hold the key down for a time equivalent to a delay, release the number, and pause for another delay. It will repeat this process for the seven digits of the pager number and then press the “send” button. (An initial counter value of 0 keeps track of the number of keys pressed and ends when the counter reaches 6). After the “send” key has been pressed, there will be a long delay. The AVR will then go to the location where the sensor code has been stored (\$Sensor Code) to send the alpha numeric page to the user of MAPSS. The sensor code is then sent and the “end” key is pressed. The flow chart for this subsystem can be found in the Appendix B under “Flowchart for Dialing”.

After the three sub systems have been carried out, the dialer AVR reenters its “sleep” state.

3) Delay Subroutine

For the purpose of dialing and encoding a pager message, the AVR micro-controller must be programmed to hold a line active for a given period of time. 2 distinct delay subroutines have been designed through the generation of clock cycle counter loops. For example, to achieve a 0.25-second delay with a 4 MHz clock requires three nested loops. Three local variables contained in registers “fine”, “medium” and “coarse” are used for the loop. The fine and medium counters run the maximum of 255 times with the coarse counter set to 8, giving just over 0.25-second delay. The flowchart is shown in Appendix B under “Flowchart for Delay Subroutine”.

4) Short Time Delay Subroutine

This short delay is required when changing the port B I/O configuration to allow time for the pin values to settle. The routine uses the global scratch register “temp” as a single loop counter for the FOR loop, set at maximum 255 passes. This provides a delay of 0.129 ms at 4 MHz. This Value could be shortened by experimentation if time is of the essence or the pins are set high prior to reconfiguration to speed things up. This might remove the need for this delay completely.

Custom Power Supply

There are three main purposes for using a custom power supply:

- 1) Source V_{cc} to all sub-components
- 2) Provide a short circuit protection
- 3) Provide a back up battery supply for the MAPSS

Figure 17 shows the schematics of this power supply.

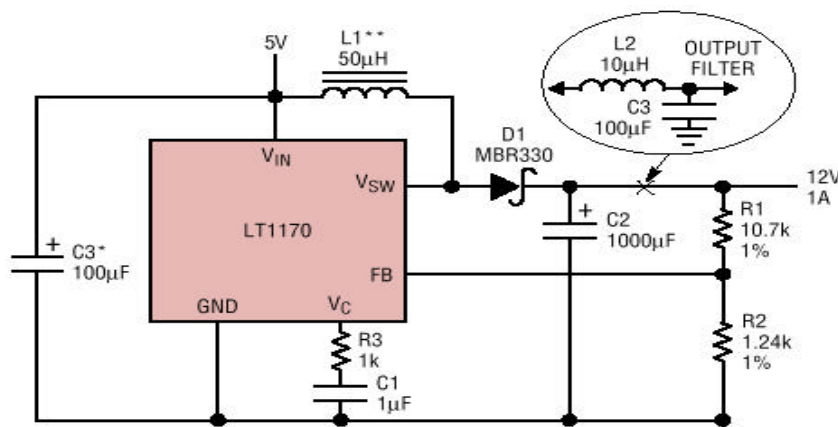


Figure 17: Custom Power Supply

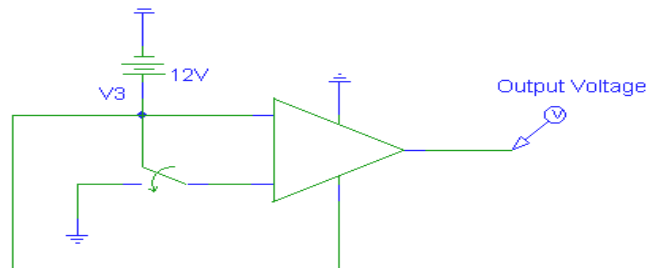
The input of the power supply comes from the 12V-car battery and the output is regulated to 5V. Table 4 shows the details of all component values used in the above circuit.

Table 4: Values of All Components Required for the Power Supply

Component Descriptions	Values
1c1 , 1c2	LT 1170
J1 , J2	3-Terminal Block
JP1 , JP2 , JP3 , JP4	Wire Jumpers
L1 , L2	50 μ H
D1 , D2	Schotchy Diode (MBR 1045)
C1 , C2	100 μ F with 0.1" Lead Spacing
C3 , C4	1 μ F with 0.1" Lead Spacing
C5 , C6	1000 μ F with 0.2" Lead Spacing
R1 , R2	1 K Ω Miniature Metal Film Resistors
R3 , R4 , R5 , R6	Different Resistor Values Depending on the Output

For the purposes of this project, a LT1170 acts as a boost converter and it is used like a switching regulator; therefore, even with unregulated input voltages, the output voltage remains regulated between the range of 3V to 60V. This unit is self-protected against any overloads and it can be externally synchronized. It draws only 6mA quiescent current and it delivers a load power up to 100W with no external power devices. By utilizing current-mode-switching techniques, it provides an excellent AC and DC load and line regulation.

In order to power the MAPSS system, the input of the power supply is connected to the car battery. However, in order to enable / disable the MAPSS system a switch is used before the power supply. Since the user might receive a lot of current from the car once the switch is on, a circuit protection is used before the switch to protect the user from current overload. Figure 18 shows the circuit protection schematic for the user protection.


Figure 18: Circuit Protection Schematics

If the switch is in the on position, then the output voltage is 12V. If the switch is off or shorted, then the output voltage is zero. Another possible solution to this is shown in Figure 19.

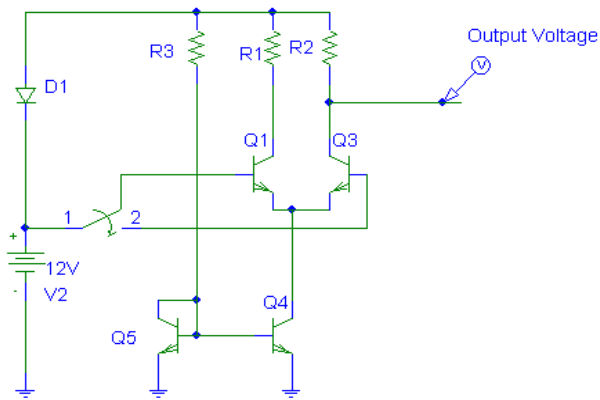


Figure 19: Circuit Protection Schematics

Testing Algorithm

1) Programmable User Interface

The user interface subsystem will be tested using the following procedure.

1. Connect a digital scope to the UART-out pin of the keypad AVR.
2. Verify proper asynchronous signal is output upon key press.
3. Generate UART signal using digital function generator.
4. Feed signal into LCD module PIC, verify signal output on LCD screen.

2) Sensor Input and Encoding

The Headlight signal sensor will be tested using the following procedure. (Component names referenced from figure 8)

Typical Case:

1. Input a 12V supply to simulate the signal from headlight and a current of 4.3A.
2. Measure the Vout signal and make sure that the voltage is at 5V.
3. Measure the current through R2. Since we know that the current coming from the headlight is 4.3A, we can determine the current that would be going into the micro-controller through Vout. The current should not exceed 20mA.
4. Input a voltage of 0V at the input and determine the value of Vout. Make sure it is at 0V.

Sometimes the value of the car battery can exceed that of 12V.

1. Input a value of 14V and determine Vout. It should not exceed 5V due to zener diode.
2. Measure the current value as that of the normal case and make sure the current going into the microcontroller does not exceed 20mA

The ignition signal sensor will be tested using the following procedure. (Component names referenced from figure 9)

Typical Case:

1. Measure the value of Vout of the above circuit with V1 at 12V and ensure voltage is 0V.
2. Measure the Voltage of V+. It should be at about 5V.
3. Decrease the value of V1 by 3V and measure the value at Vout. The output should be at 5V.
4. Place a large Rload at the output and measure the currently going through Rload. This value should not exceed 20mA.
5. While measuring the output, determine the time length at which the output is at 5V. The time should be about 100ms long so that the signal can be detected properly.

Extreme Case:

1. Decrease the value of V1 at a faster rate and determine the time length at which Vout is at 5V.
2. Repeat the above instruction many times at a faster rate.

The encoder used for the signal sensing will be tested using the following procedure.

1. Replace the microcontroller with a voltage source. Vary the input signal and use the voltage source as select signals and then measure the Enable signal to see whether the logic is correct.
2. Input a 5V into the alarm line and use 0V for both s0 and s1 to select it. Measure the Enable and check to see if it is between 3.7-5V (HI).
3. Use the same procedure to test all other signals of the MUX.
4. Replace the voltage source with the AVR microcontroller and test the signals again as mentioned in the previously.

3) Page Encoding and Dialing

The page encoding and dialing unit will be tested using the following procedure.

1. Generate UART signal using digital function generator.
2. Feed in 7 digit number into \$phone memory location.
3. Place a valid TTL sensor code on transmit bus.
4. Activate low interrupt on dialer AVR chip.
5. Place weighted color LED's on output bus, and verify output matches phone number and sensor code.

4) Overall System

The entire MAPPS system will be tested using the following procedure.

1. Turn off car.
2. Turn on MAPPS power switch.
3. Enter 7 digit pager number using keypad.
4. Enable car alarm system.
5. Kick car, impact alarm momentarily "chirps", MAPPS system delivers alphanumeric page "I".
6. Attempt opening door, alarm activates, MAPPS system delivers alphanumeric page "A".
7. Turn car on, MAPPS system delivers alphanumeric page "G".
8. Repeat steps 1-3.
9. Turn car lights on for 15 minutes, MAPPS system delivers alphanumeric page "L".

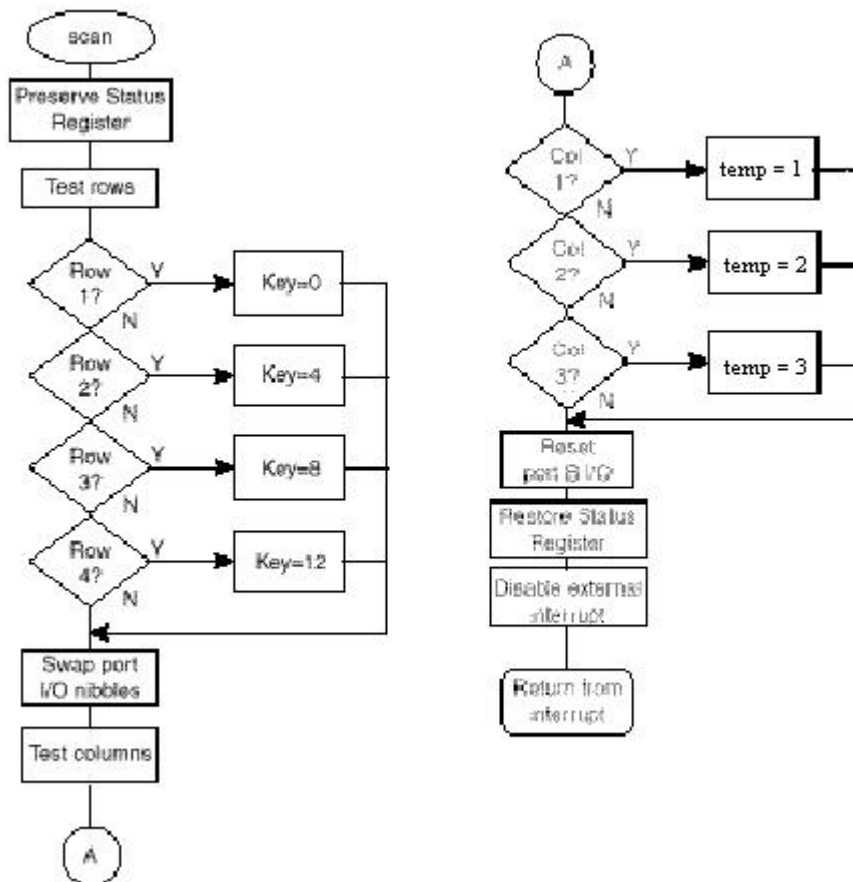
References

- [1] Atmel Corporation, *AT890s Microcontroller Design Kit User's Guide*, 1997.
- [2] Hitachi Incorporated, *HDT6654c Liquid Crystal Display Data Sheet*
- [3] Mano, Moris. *Computer System Architecture*, 3rd ed. Prentice Hall, Englewood Cliffs: 1997.
- [4] Microchip Corporation, *PIC16C Data Sheet*, 1996.
- [5] Motorola Corporation, *Motorola Flip Phone 650 User's Manual*, Washington: 1995.
- [6] Phillips, Harbor. *Feedback Control Systems*, 3rd ed. Prentice Hall, New Jersey:1996.
- [7] Sedra and Smith, *Microelectronic Circuits*, 4th edition, Oxford University Press: 1997
- [8] Toyota Corporation, *Toyota Paseo User's Manual*

Appendix

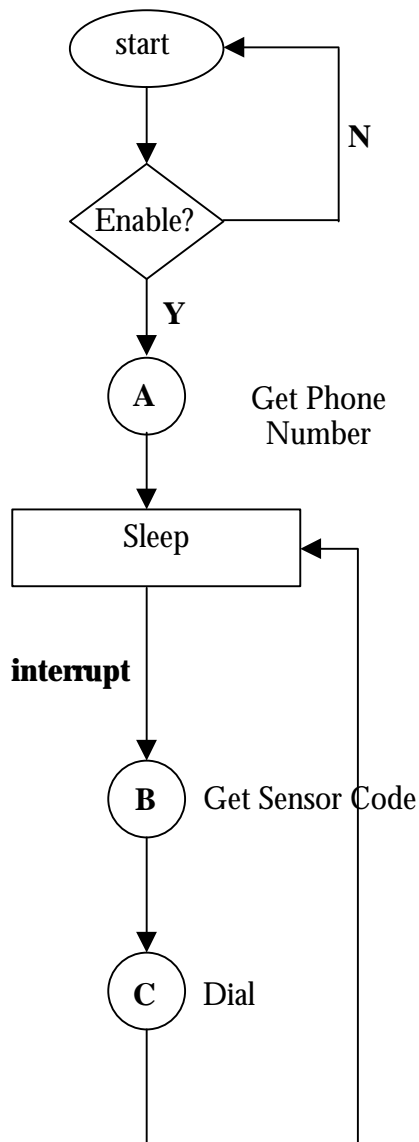
A - Programmable User Interface

Flowchart for Interrupt Service Routine

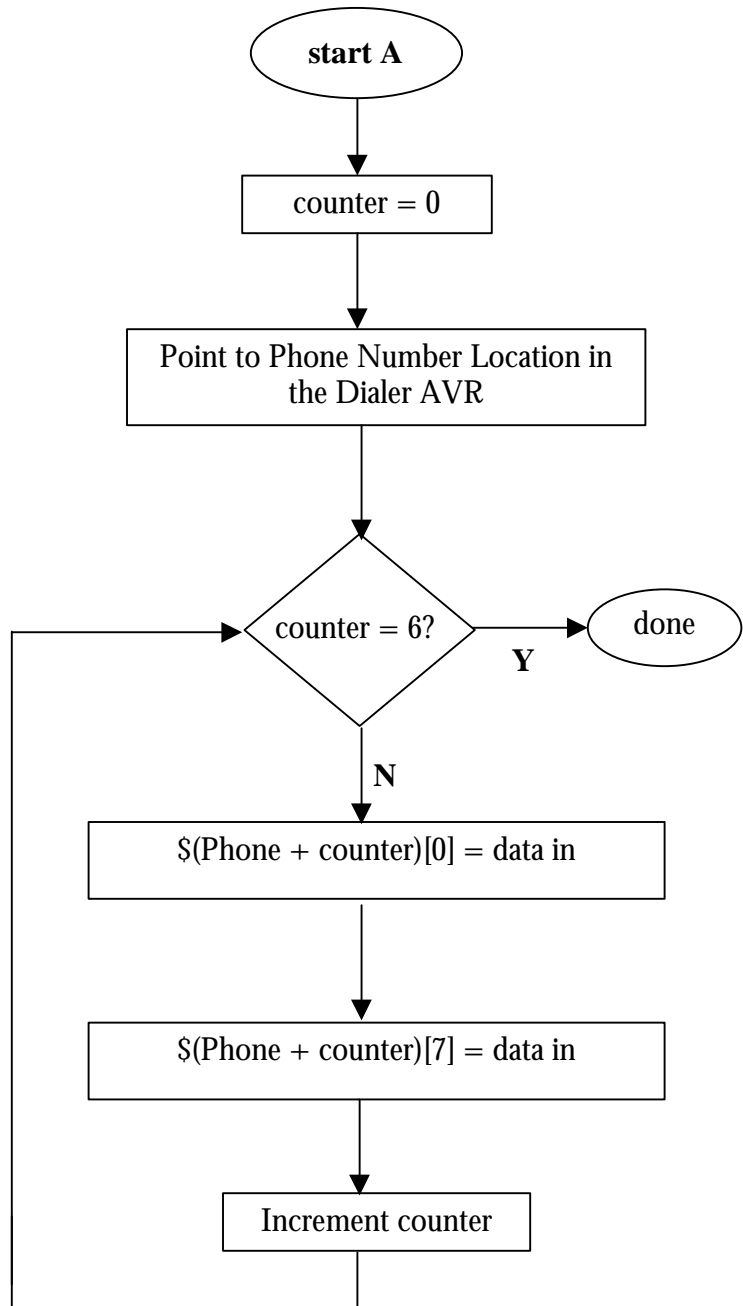


B - Page Encoding and Dialing Unit

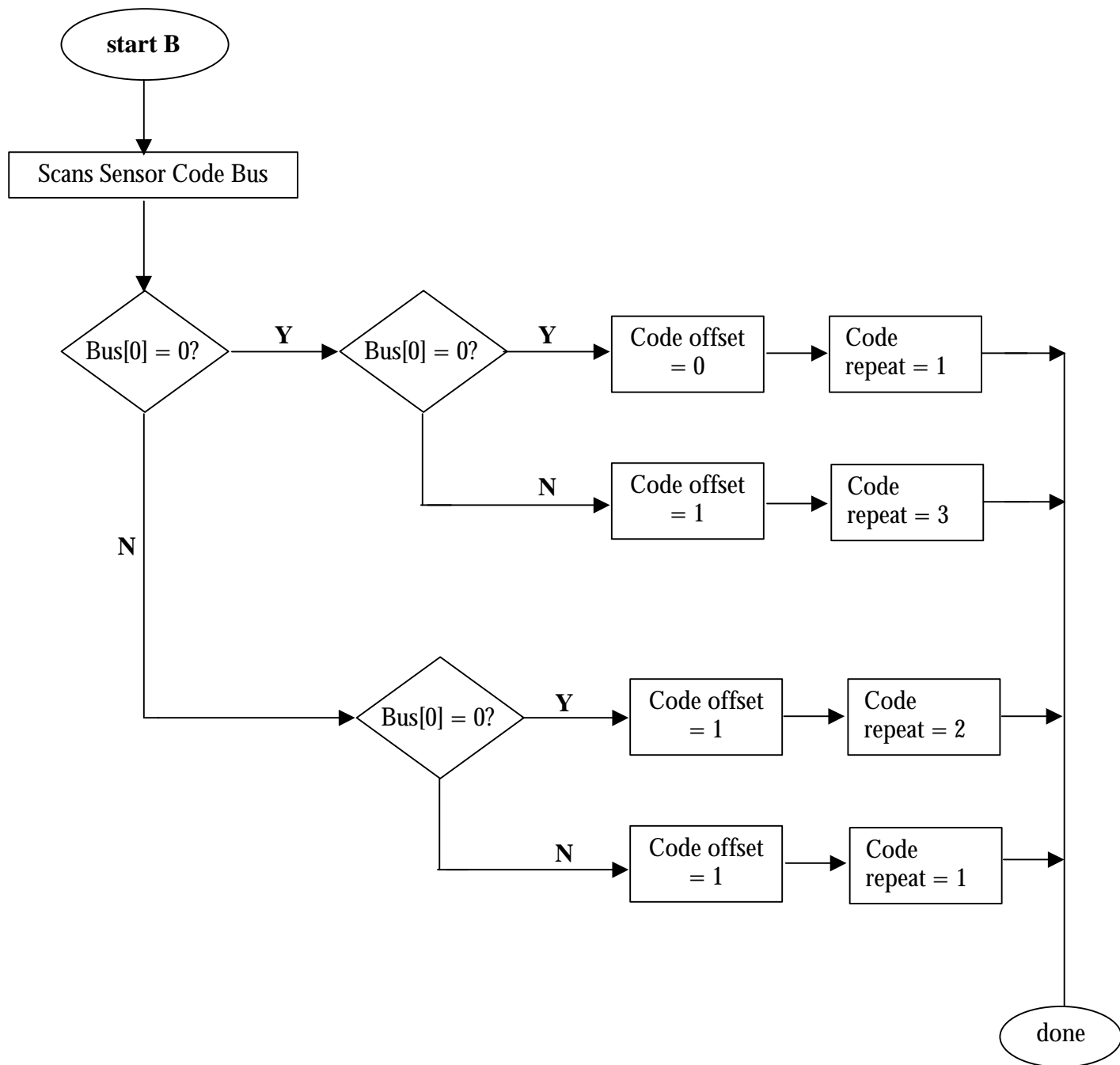
Flowchart for Main System in the User Interface Subunit



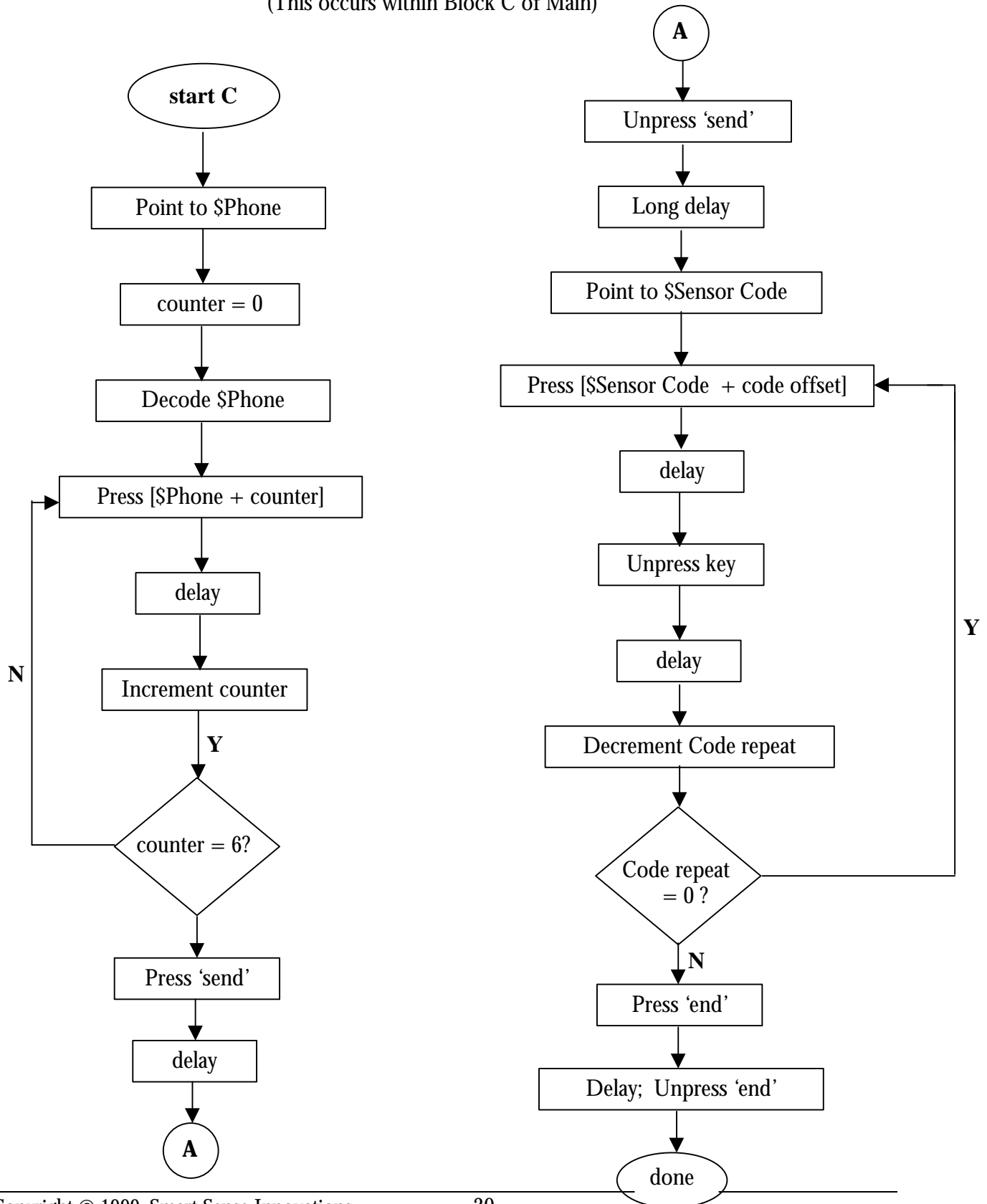
Flowchart for Retrieving the Phone Number from the Keypad AVR
(This occurs within Block A of Main)



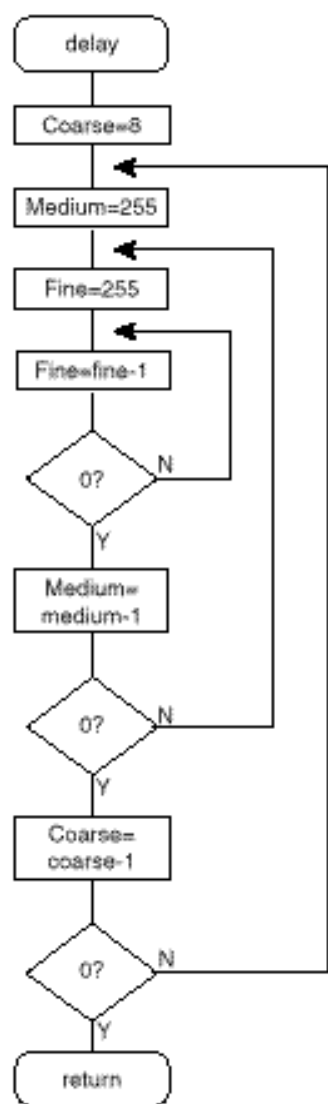
Flowchart for Retrieving the Sensor Code from the Keypad AVR
(This occurs within Block B of Main)



Flowchart for Dialing
(This occurs within Block C of Main)



Flowchart for Delay Subroutine



Flowchart for Transmit Subroutine

