Real-Time Vehicle Communication

Supervisor:
Dr. Peter Driessen

Prepared By:
Christian Pedersen
Bemnet Lulie
Chol Nhial

Executive Summary

Real-Time Vehicle Communication uses vehicle CAN communication, BPSK transmitter and receiver, and microcontroller to notify vehicle owner of possible theft in real-time. This device monitors the vehicle's ECU data to determine the state of the vehicle. The necessary ECU data is gathered and analyzed by the device via CAN bus, the vehicle's main communication bus. Once the device determines the vehicle is stolen, it uses a BPSK transmitter to send an HF signal at a 2.953 GHz frequency to a BPSK receiver. The receiver feeds the transmitted signal to a microcontroller by translating the HF signal to a digital signal. The microcontroller uses the incoming digital signal as a trigger for flashing an LED light to alert owner.

Table of Contents

1 Introduction

Real-Time Vehicle Communication (RTVC) is a one-way security communication device that alerts the vehicle owner about possible thefts. This will allow the vehicle owner to notify the police sooner, decreasing police response time.

The RTVC package consists of two devices: a transmitter and receiver. The transmitter is installed into the vehicle connecting to the vehicle's communication network called controller area network (CAN). The transmitter decodes the necessary CAN signals to determine whether the vehicle is being stolen. Once the device realizes the vehicle is being stolen, it will transmit warning signal to the receiver through the vehicles antennae. The receiver is a small portable device that can be carried from a keychain and flashes an LED to alert the vehicle owner of possible thefts.

This device will be useful for all vehicle owners, providing extra security to limit theft.

2 Problem

Vehicle recovery rates in Canada have been dropping over the past couple decades. Average recovery rates in Ontario have dropped from a 90% recovery rate in 1990 to about 50% in 2010 [2]. With the decrease in recovery rates, vehicle owners are looking for secure ways to protect their vehicles.

When electronic security devices are inserted into vehicles, the recovery rate increases drastically. When recovery rates from vehicles with security devices were compared to regular vehicles in the states, cars with security devices had a 90% recovery rate. This recovery rate almost doubles the no security recovery rate of 43.2% [3].

The decrease in vehicle recovery rate has given rise to many security companies like OnStar. OnStar, however, is only available for GM vehicle users and require monthly subscription. Therefore, RTVC hopes to cut out the middleman and alert the vehicle owner to potential thefts and reduce the response time. By providing a direct connection from the vehicle to the owner, the device will not require any monthly fees. This will decrease the security cost into a one-time fee.

RTVC hopes to provide a secure, inexpensive system that alerts vehicle owners to potential thefts.

3 Solution

In order to decrease police response time, RTVC introduces a security device that alerts the user about potential thefts. The security device will transmit decoded internal CAN signals from the vehicle, and a separate portable device receives the signal and alerts the vehicle owner with a flashing LED.

RTVC uses four devices to transmit and receive decoded CAN signals: CAN Decoder, BPSK Transmitter, BPSK Receiver, and a Microcontroller. Figure 1 shown on the next page displays the complete flowchart of the RTVC package

.

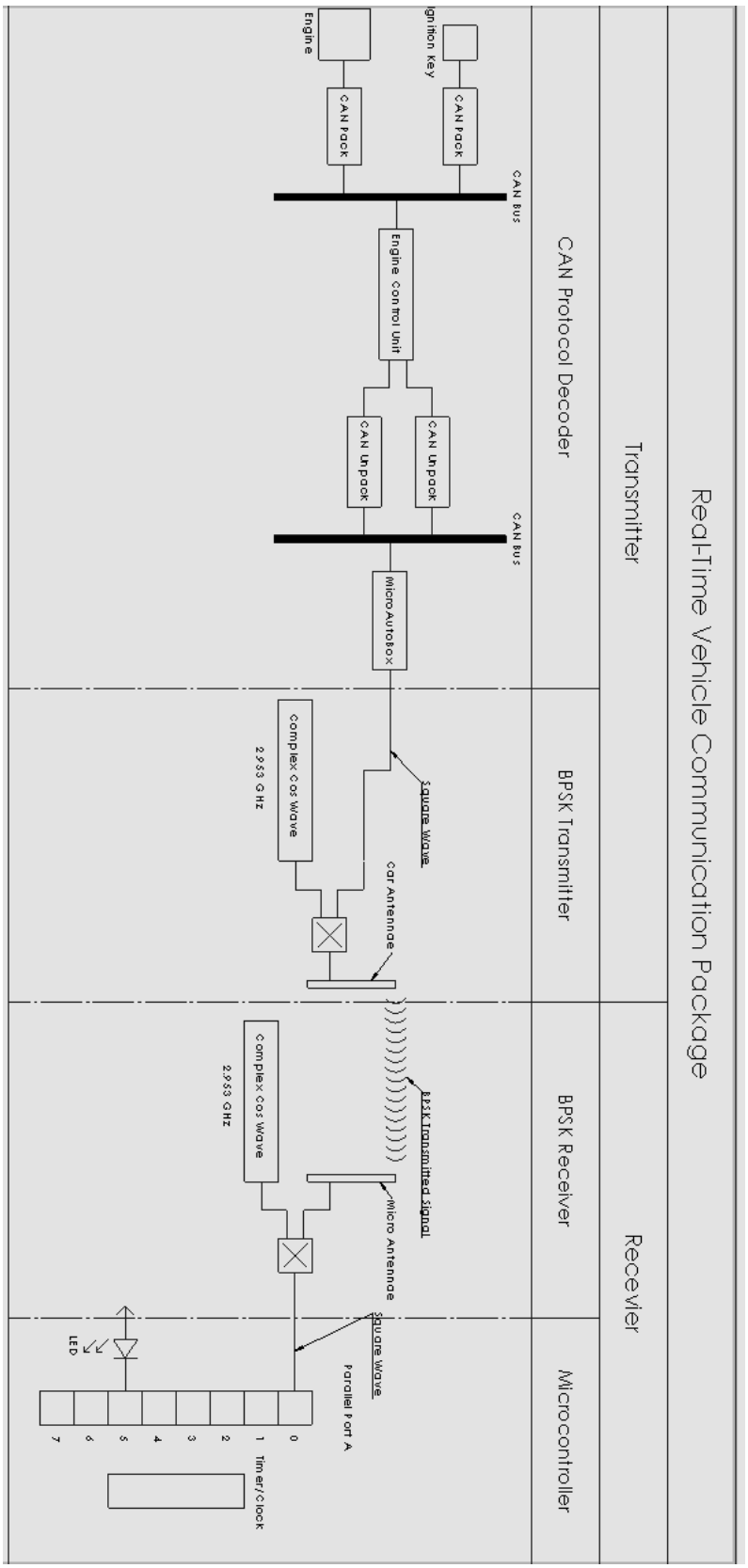

Figure 1: Real-Time Vehicle Communication Package Flowchart

3.1 CAN Decoder

3.1.1 CAN Communication

Controller Area Network (CAN) is an automotive industry standard for communication between several vehicle electronic control units (ECUs). Each ECU is connected on a shared CAN bus that is able to transmit and receive messages from the network. CAN bus is based on the "broadcast communication mechanism" [13], meaning every ECU on the CAN bus is capable of accessing every message transmitted on the network. However, the ECU uses filters to receive specific messages from the CAN bus. For the ECU to recognize the desired message, every message has an identifier. The message identifier allows CAN messages to be unique from the whole network. This is important because each identifier message defines specific messages and also priority of the messages.

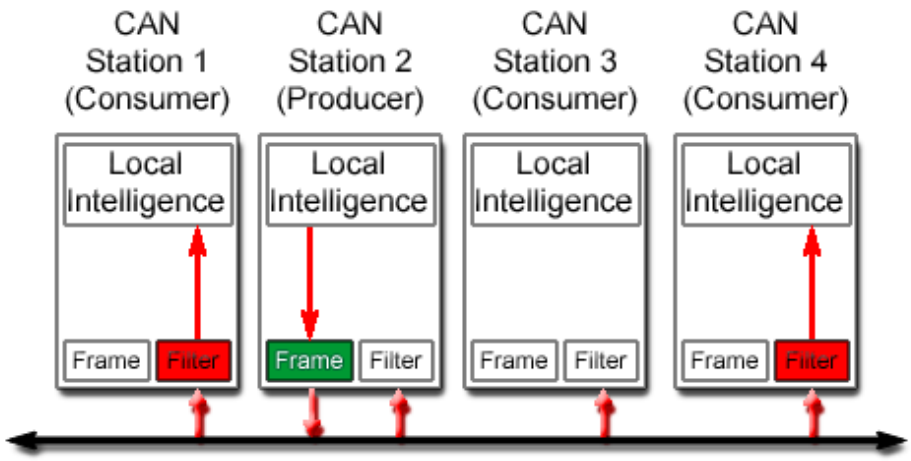

Figure 2: CAN Communication between CAN Stations or Nodes [13]

Figure 2 illustrates the interaction between several CAN stations (nodes). CAN stations (nodes) comprise of ECUs and devices that transmit and receive message to and from the CAN bus. However, the ability for each node to transmit on the same CAN bus leads to bus access conflict. Therefore, messages use bit-wise arbitration and priority identification to manage bus access. A node with higher priority will "win" the

arbitration to transmit, hence being the “Producer” node, while the rest become the “Consumer” nodes.

Not all “Consumer” nodes accept the transmitted message. By filtering the message type that is transmitted, the “Consumer” node can decide to accept or deny the message. The “Producer” node uses frame to specify the message type.

CAN standard comprise of four types of message frames:

Data frame: Consists of node data that transmits message
Remote frame: Consists of node data that requests a specific message
Error frame: Consists of node data that transmits error flag when error is detected
Overload frame: Consists of node data that governs transmission overload between Data and Remote frames

3.1.2 CAN Decoder Application

Specific CAN messages will be tracked to detect whether the vehicle has been compromised. CAN messages that are of interest are the ignition system and engine system. These systems are ECUs in the vehicle that are part of the CAN bus. By adding a CAN decoder device on the CAN bus which requests message from the ignition and engine systems, the device will output warning if the vehicle is compromised. Figure 3 in the next page shows messages and decision process used by the device.

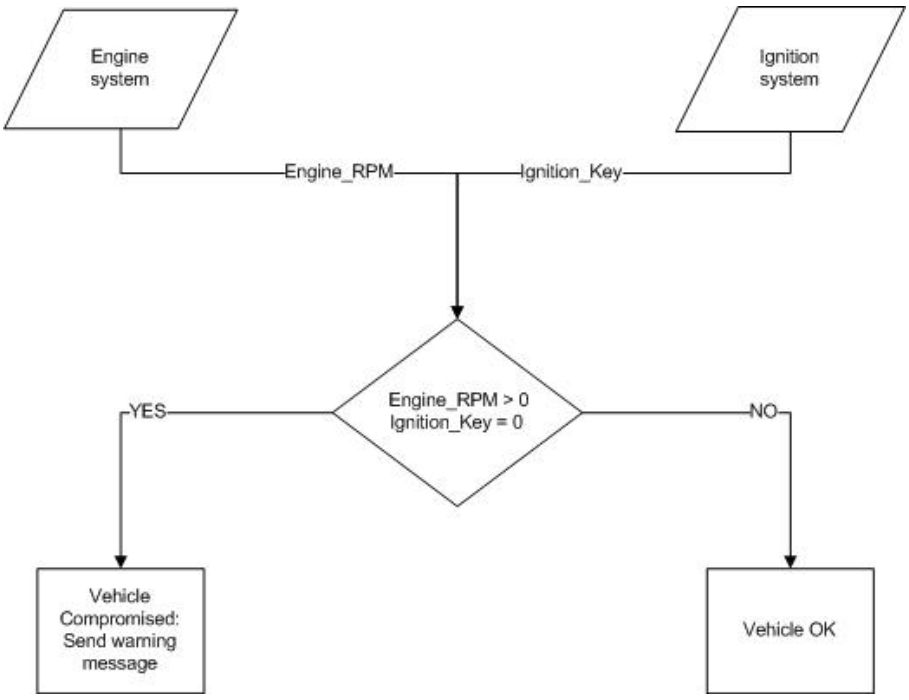

Figure 3: Device Control Logic

The device checks to see if the Ignition_Key is set to 1 when Engine_RPM is greater than 0. This logic identifies whether the engine is turned-on using the ignition key. If the device detects that the Ignition_key is still 0, while the Engine_RPM is greater than 0, it identifies the engine being turned-on improperly. Hence, the device goes to "Vehicle Compromised" mode to send warning for the user. The warning consists of sending a 500 Hz square wave to the BPSK transmitter.

3.2 BPSK Transmitter

The BPSK Transmitter is a device that converts the digital square wave output from the CAN Decoder into a physical signal, and transmits the signal using the vehicle's antennae. The software used to simulate the transmitter and receiver is GNURadioCompanion (GRC). This software simulates real components to produce transmitters and receivers.

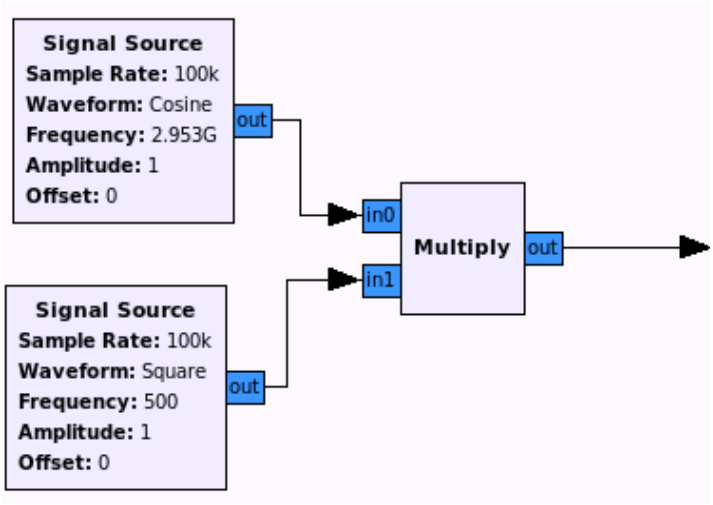

Figure 4: BPSK Transmitter using GNU Radio Companion (GRC)

To convert the square wave to a transmitted signal, the transmitter multiplies the input square wave by a complex cosine wave of 2.953 GHz as shown in Figure 4. This produces a transmitted signal, which is then broadcasted through the vehicles antennae. The sampling rate was chosen to be 100k, so as to create a square wave with 1.67 kHz frequency when the square wave is received. Figure 5 shows the transmitted signal.

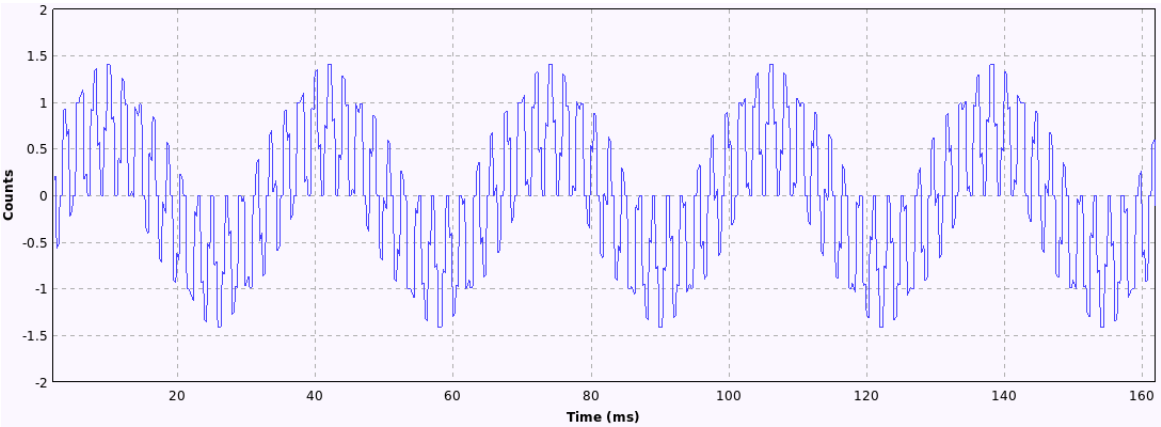

Figure 5: BPSK Transmitted Signal

$$m(t) = \ square\ wave; f = 2.953\ GHz$$
$$Transmitter \rightarrow m(t) * e^{2\pi ft} = \ m(t)e^{2\pi ft}$$

Equation 1: BPSK Transmitter Signal Equation

In order to create a signal that will transmit over longer distances, we up converted the signal to a higher frequency. This frequency depends on the length of the receiver antenna. Based on the average distance a person walks to take transit, we will use 400m as the average distance a person is willing to travel away from their vehicle [6]. The obstacles covered would be a cityscape with buildings, vehicles, and noise all contributing to radio interference.

The carrier frequency chosen is 2.953 GHz, as a 1" antenna can accept this frequency. At this frequency the signal would experience minimal loss due to interference and noise. The maximum receiver distance d was chosen to be 1.8 km. Since we do not expect vehicle owners to travel more than 400m from their vehicle, this provides an appropriate transfer range. The below calculations show the frequency, receiver gain and transmitter gain. The calculations are based on an antennae transmitter gain of 4 dB [9] and a receiver gain of 0 dB (≈0) [10].

$$antenna\ receiver\ length\ (arl) = \frac{\lambda}{4} = \frac{c}{4 * f} \rightarrow$$

$$\frac{c}{arl * 4} = \frac{3 * 10^8}{0.0254 * 4} = 2.953\ GHz\ frequency$$

So the frequency the BPSK transmitter needs to convert the square wave by is 2.953 GHz. So we use this frequency in the L_O calculation.

$$P_R = P_T + G_T + G_R - L_O$$

Where P_R is received power, P_T is transmitted power, G_T is transmitting antennae gain, GR is receiving antennae gain, and Lo is free space loss. Since P_T is controlled by an amplifier in the BPSK Transmitter, we set L_O when d = 1.8 km to simulate the worst-case free space loss.

$$L_O = 20 \log\left(\frac{\lambda}{4\pi d}\right)$$

Where d is distance (km), λ is the wavelength (m) of the transmitted signal. Since d = 1.8 km and λ = 0.1016m, the worst-case free space loss can be found.

$$\lambda = .1016\ m$$

$$L_O = 20\log\left(\frac{\lambda}{4\pi d}\right) = 20 * \log\left(\frac{0.1016}{1.8 * 4 * \pi}\right) = -46.95\ dB$$

Equation 2: BPSK Distance Equations

GT and GR are 4 dB and 0 dB respectively, so the equation is:

$$P_R = P_T + 4 + 0 - 46.95$$

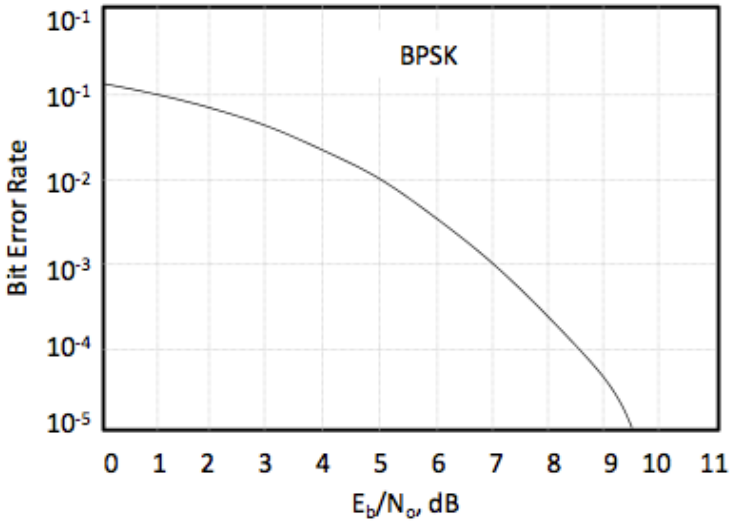

Figure 6: BPSK Gain Requirement for Bit Error

Since PR needs gain when the signal hits the receiver, the minimum gain required for P_R is ≈1 dB. This is based on the Bit Error Rate Vs P_R figure display on Figure 6. Figuring out the minimum required Bit Error Rate will require testing from the microcontroller, so we will assume a value of 1 dB required to be accepted for a Bit Error Rate of 10^{-1}.

$$1 = P_T + 4 + 0 - 46.95$$

$$43.95\ dB = P_T$$

Since the value for P_T = 43.95 dB, the BPSK transmitter will have to amplify the transmitted signal by 43.95 dB.

The maximum receiver distance d_{max} = 1.8 km, which is greater than the average distance a person travels for transit, 400 m. Therefore, our RTVC Package can operate under a 2.953 GHz frequency.

3.3 BPSK Receiver

The BPSK receiver decodes the transmitted BPSK signal, and reverts the signal back into a square wave, and sends the square wave into the microcontroller.

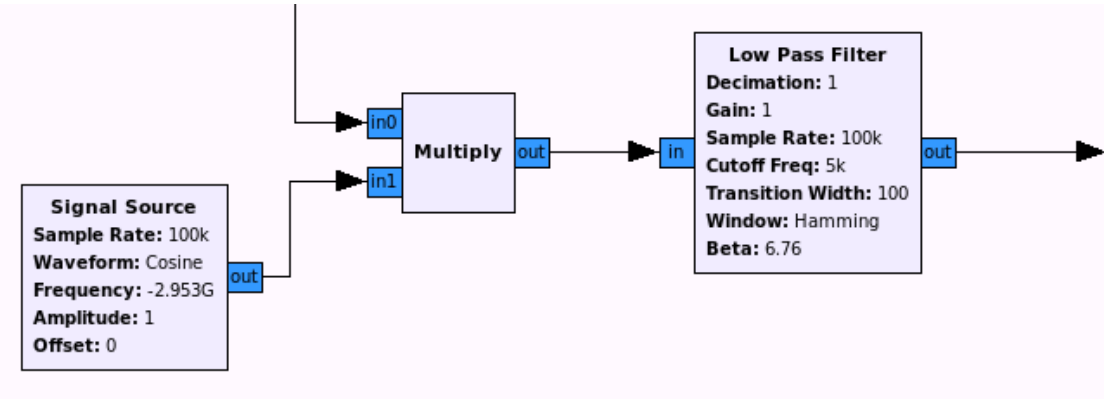

Figure 7: BPSK Receiver using GRC

The received signal enters the receiver at port *in0* of the multiplier as shown in Figure 7. In order to decode the signal, the BPSK receiver multiplies the input signal by a complex cosine wave of frequency -2.953 GHz. This frequency is used to cancel out BPSK's cosine wave of 2.953 GHz producing the original square wave. This is shown using the below equation.

$$m(t) = \ square\ wave; f = 2.953\ GHz$$
$$Receiver \rightarrow (m(t)e^{2\pi ft}) * e^{-2\pi ft} = \ m(t)e^{2\pi ft - 2\pi ft} = m(t)e^{0} = \ m(t)$$

Equation 3: BPSK Receiver Signal Equation

The original square wave (m(t)) has been recovered. The recovered square wave is displayed in Figure 8.

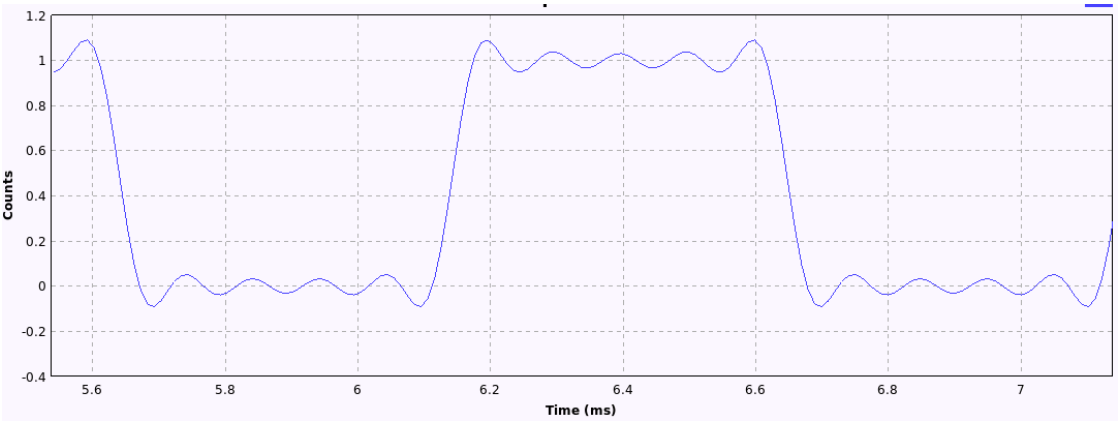

Figure 8: BPSK Down-Converted Square Wave

The output produced by the BPSK receiver resembles a square wave, however the peaks of the square signal aren't flat, retaining waves from the cosine signal. The original signal is impossible to recover; however with the addition of RX filters and amplifiers the noise can be reduced.

3.4 Microcontroller

The Microcontroller device used in the Real-Time Vehicle Communication package accepts the decoded square wave signal, and causes the built in LED to flash repeatedly.

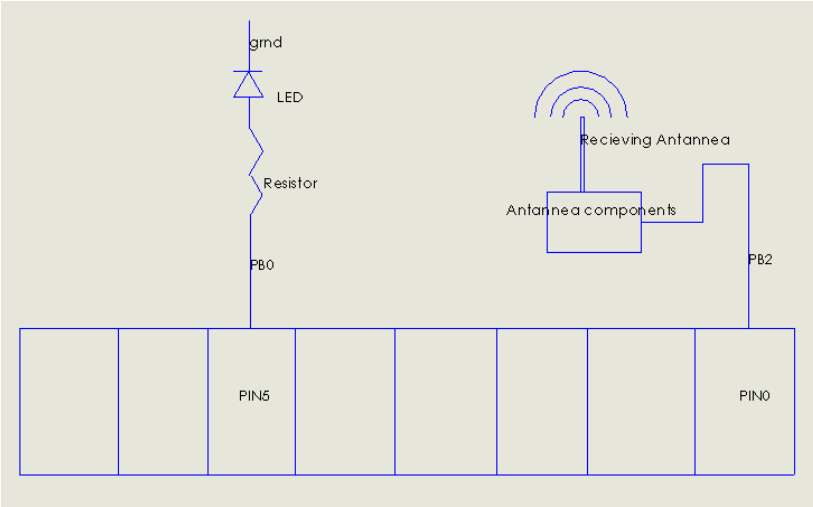

Figure 9: Designed Microcontroller

Figure 9 shows a proposed microcontroller design. The microcontroller uses its parallel ports to input the square wave (port 0), and to output power to the LED (port 5). This is done using internal interrupts of the microcontroller; a parallel port interrupt, and a timer interrupt. Comments are shown after a *%*.

3.4.1 Parallel Port Interrupt

```
void Intwarn_Init()
{
*PADIR = 0x20;                          % LED output, BPSK receiver input
*IVECT = (volatile int*) &Intwarn;   % set up interrupt vector
asm("MoveControl PSR, #0x40");   % CPU responds to Interrupt
*PCONT = 0x10;                          % Enables port A interrupts
}

_interrupt_void Intwarn_handler()
{
if((*PBIN & 0x1) != 0)                  % When a signal is received
{
*CTCON = 0x11;                          % Begin Timer so LED flashes
}
}
```

Figure 10: Parallel Port Interrupt Initialization and Handler

To initialize the parallel port interrupt, we use the function *Intwarn_Init()* to set up the parameters of the interrupt. *Intwarn_Init()* first assigns the LED port to output using the register **PADIR*. This allows the parallel port to output electricity to the LED. Since the LED is located at the 5th port, we set **PADIR* as the hexadecimal digit for port 5, 0x20 or 0010 0000 (0 = input, 1 = output). Next, the code sets up the initialization with the interrupt *Intwarn_handler*. This is done by assigning the initialization to *Intwarn* using the register **IVECT*. The code involving *asm()* enables the CPU to respond to the

vectored interrupt. Finally, the initialization enables parallel port interrupts, which means any input into port 5 of parallel port A will initiate the interrupt.

Since the LED is supposed to flash as soon as the receiver receives the signal, we want the input signal to generate the LED. *Intwarn_handler*() begins with the input warning signal. The *if* statement checks if the port causing the interrupt is port 5. This is necessary, as it eliminates the possibility of shorts or noise generating the interrupt from other ports. After the check the interrupt begins the timer, enables the timer interrupt, and exits *Intwarn_handler*(). Figure 10 shows the interrupt initialization and handler, and Figure 12 displays the registers used.

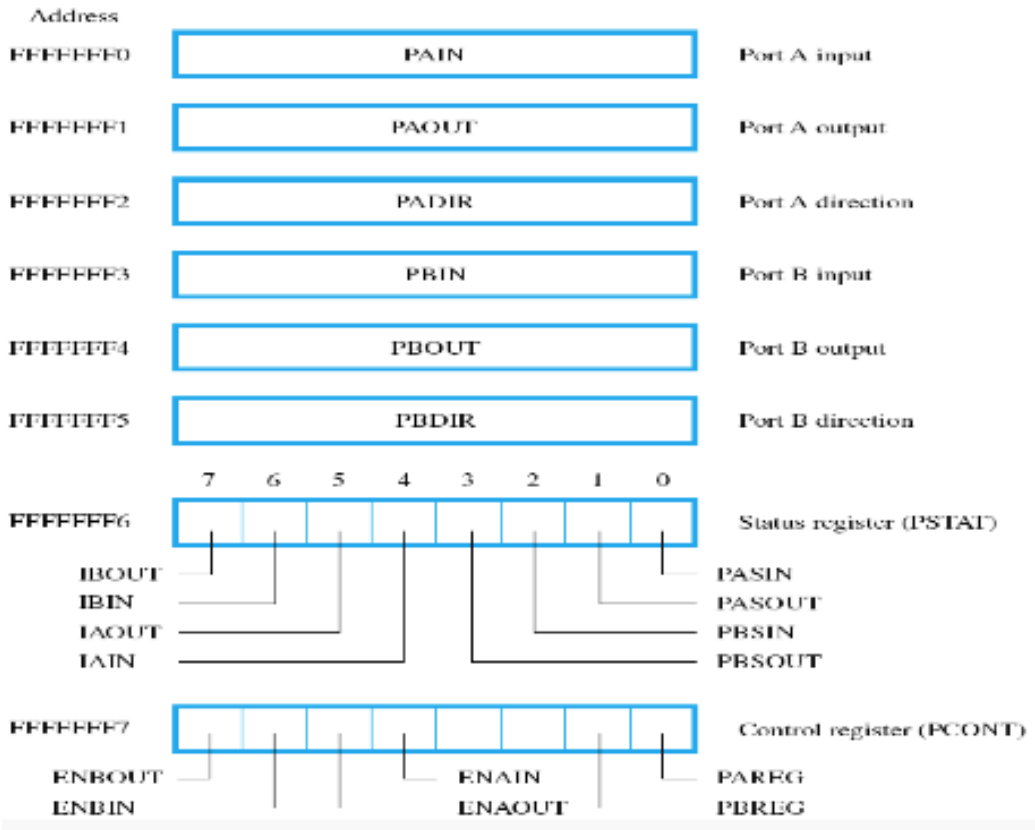

Figure 11: Parallel Port Registers [1]

3.4.2 Timer Interrupt

```
void Timer_Init(void)
{
*CNTM = 50000000;        % Sets timer to 1/2 second
*CTCON = 0x10;           % Enable Timer interrupts
}


_interrupt_void Timer_handler()     % led flashes every half second
{
if(LED == 0x0)
{
LED = 0x20;
}
else
{
LED = 0x0;
}
*PAOUT = LED;           % Update output
*CTSTAT = 0x0;          % Clear "reached 0" flag
}
```

Figure 12: Timer Interrupt Initialization and Handler

The code begins with the initialization of the timer interrupt through the function *Timer_Init(void)*. *Timer_Init(void)* sets the timer clock **CNTM* to count to 0.5 seconds. Then, the timer interrupt is turned on, however the timer is not started. The timer interrupt causes the computer to enter *Timer_handler*() eveytime 0.5 seconds is counted to by the timer. The code used to start the timer is in the parallel interrupt *Intwarn_handler*().

Timer_handler() executes the timer interrupt every 0.5 seconds after the timer is enabled in *Intwarn_handler*(). The timer handler checks to see if the LED is on. If it is on (LED == 0) then the handler turns it off (LED = 0x20). If the LED is off the handler turns it on. Finally, the code updates the output to parallel port A (**PAOUT*) to LED, so that the LED will turn off or on; and the timer flag is reset to exit the handler. After 0.5 seconds, the handler will execute again. Figure 13 shows the registers used.

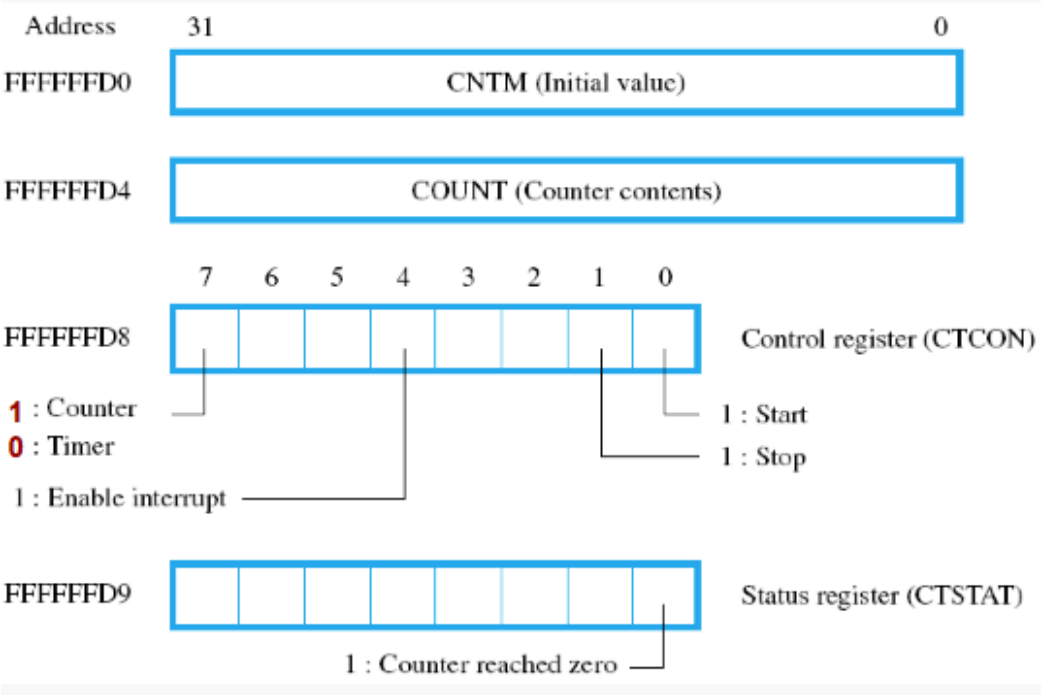

Figure 13: Timer Register [1]

The code used in the above section uses pseudo code to simulate a microcontroller.

4 Design Implementation

4.1 CAN Decoder

dSPACE midsize simulator will be used to simulate vehicle signals that would be connected to the decoder via CAN bus. The simulator is comprised of real-time processer, I/O board and CAN bus that allows for development and testing of ECUs and devices. dSPACE ControlDesk software will be used test the device for conditions that was discussed in3 Solution3.1.2 CAN Decoder Application. ControlDesk is experimental software that uses graphical interface to write and read vehicle signals to and from the simulator.

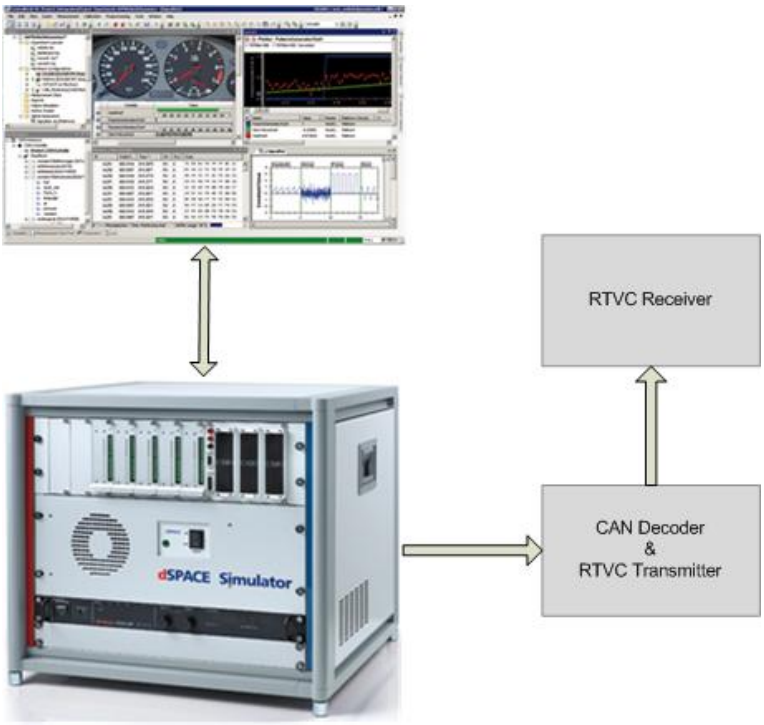

Figure 14: Design Implementation via Simulation

Figure 14 shows the interaction between the ControlDesk software, the simulator and the decoder that receives the necessary CAN messages to broadcast to the user via BPSK transmitter if the vehicle is compromised. The decoder will use MCP2551 chip to connect with the CAN bus to extract CAN messages from the simulator. The extracted

CAN messages will be sent to dsPIC30F4012 chip, which identifies and decodes the necessary CAN messages. Detailed specifications of MCP2551 and dsPIC30F4012 chips are found in [14] and [15] respectively. Once the dsPIC30F4012 chip detects that the vehicle has been compromised based on the control logic shown in section 3.1.2, it will output square wave signal. The square wave signal is feed to the BPSK transmitter.

4.2 BPSK Transmitter

The BPSK transmitter can be created using digital to analog converters, mixers, quadrature oscillators, and a combiner. Figure 12 below shows the transmitter design.

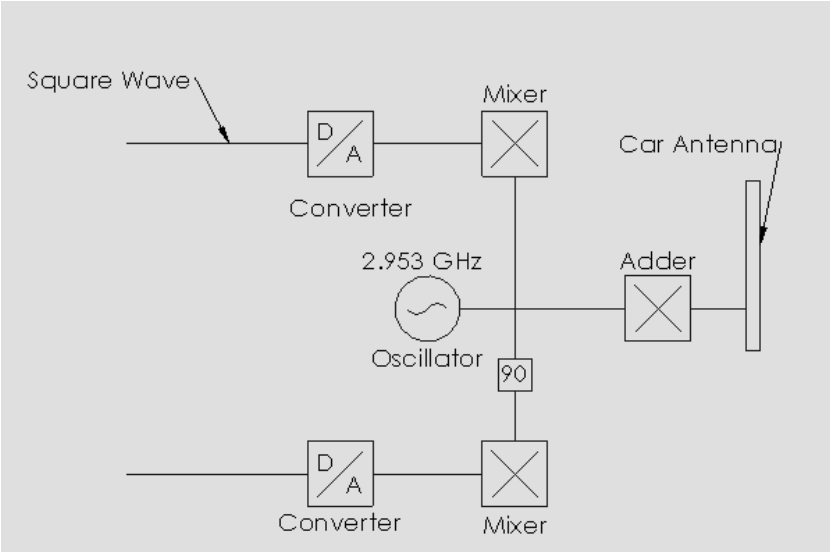

Figure 15: BPSK Transmitter Design

First, the real signal would pass through the D/A converter, converting the digital square wave into analog data. The imaginary signal would be connected to ground. An internal 2.953 GHz cosine wave created by a quadrature oscillator is multiplied by the analog data to produce a physical signal. The 2.953 GHz cosine wave is also phase shifted by 90˚ to create a 2.953 GHz sine wave, which is multiplied by the imaginary signal. To multiply the data with the generated signals, the BPSK transmitter can use double

balance mixers. Finally, the real and imaginary signals are added together using a combiner and are transmitted through the vehicles antennae [12].

Building a transmitter will take an estimated 6 hours with the building materials costing $10.00. This is based on the material costs and time taken building a soft rock transceiver. When working an hourly fee of $10.00 per hour, the total cost to build a BPSK transmitter is $70.00.

The antenna used as a parameter for RTVC is the TQC-400FI-477 model vehicle antenna. The antenna has a gain of 5.5 dB, which will increase the distance that the receiver will function. The antenna costs $59.95, however will come to no extra cost to the consumer, as the vehicle already has an antenna [9].

Transmitter Cost	$70.00
Transmitter Cost (With Antenna)	$129.95

Table 1: Total BPSK Transmitter Costs

4.3 BPSK Receiver

The BPSK Receiver will be created in the same way as the BPSK Transmitter. The receiver uses a 1“ antenna, RX Filters, a low noise amplifier (LNA), automatic gain control (AGC), mixers, quadrature oscillators, and a combiner. Figure 13 displays the BPSK Receiver design.

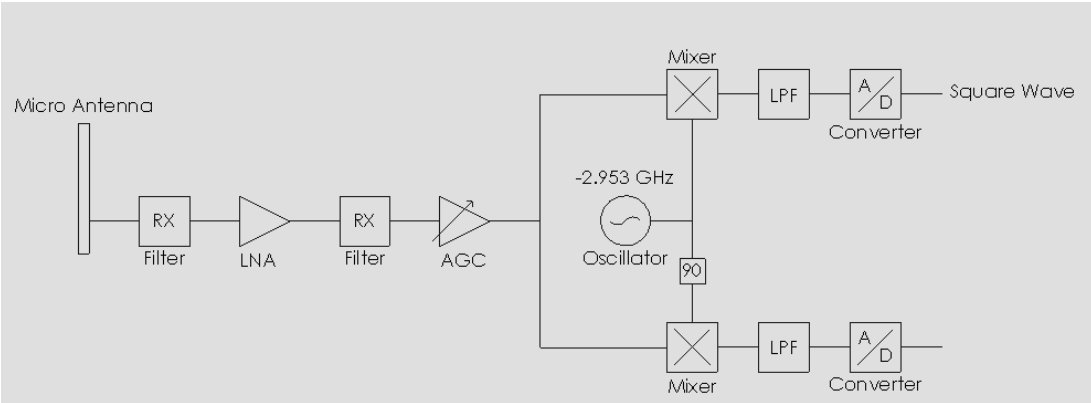

Figure 16: BPSK Receiver Design

The antenna used to receive the signal is a 1" Short Stubby" antenna. The antenna has a gain of ≈0 dB. This antenna is used in the RTVC receiver because of its smaller compact design. The antenna costs $4.97.

The receiver begins by accepting the signal through the 1" antenna. The signal passes through a RX filter to cancel out signals at similar center frequencies. The LNA amplifies the voltage of the received signal to a voltage usable by the converters. AGC varies the gain of the signal to cancel out variations in the received signal. Next, the receiver cancels out the $2.953\ \mathrm{GHz}$ cosine and sine wave by multiply the signal by a cosine and sine wave with the frequency of $-2.953\ \mathrm{GHz}$. This is done using the mixers, a 90˚ phase shift and a quadrature oscillator. Finally, LPF's are used to cancel out image frequencies and the recovered square wave is outputted [12].

We estimate that BPSK Receiver materials will cost $15.00, with a building time of 9 hours. This is based on the material costs and time taken building a soft rock transceiver. When working an hourly fee of $10.00 per hour, the total cost to build a BPSK transmitter is $105.00.

Receiver Cost	$109.97

Table 2: Total BPSK Receiver Cost

4.4 Microcontroller

4.4.1 Requirements and cost Analysis

The problem required a microcontroller that will receive data from the receiver antennae. This data will raise an interrupt for the microcontroller to switch the LED on as a form of notification to the owner of the vehicle. This function could also be done with a Liquid Crystal display LCD however the cost of the gadget will increase, as an LCD would cost between 12 to 18 dollars. Since it's an alert that is required to detect theft an LED could do the same work conveniently. Full implementation of this part of the project will need the materials presented in table 1.0. The cost for each component is

given as well. A battery to power the chip should supply 5 volts and can be dry cells or any other source of power compatible with the device volume.

16-8bit AVR Microcontroller	$8.50
1LED	$0.34
Resistor	$0.24
Total price	$9.08

Table 3: Microcontroller Cost

4.4.2 Technical implementation

The microcontroller will be supplied 5 volts through its V_{cc} or supply pin PB10 and will be grounded at pin 11 the ground. A push button will be connected through the line that feeds the V_{cc}, this will ensure that the user can switch on and off the device whenever for convenience. For example, after an alert has been noticed or the user is in the car, the controller will not need to be on. The LED that will create the alerts is connected to portB0. V_{cc} input can blow out the LED therefore a resistor is used to limit the current fed into the LED. The LED that will be used can have a rating of 2 volts and current of about 0.01. Equation 4 below can be used to calculate the value of the resistance to be used.

$$Resistance = \frac{V_{cc} - V_{LED}}{I_{LED}}$$

Equation 4: Resistance Calculation

The receiver Antennae will be connected to the microcontroller through PORTB2 which will act as a source of interrupt for the microcontroller. Figure 17 shows the pins connection of the microcontroller. Having discussed the pins layout of the most important pins of the chip to this project, it is now time to outline the functionality of the device built. The receiver antennae will get signals from the car antennae (sender). These signals are channeled to PORTB2 of the microcontroller as pulses of highs and lows. This port is used for an external interrupts and respond only if a high is detected.

The C code shown in figure 1.2 enable the microcontroller to turn the LED on and off at an interval of 100 milliseconds. Once the LED starts blinking, the user knows that the vehicle is at risk and will therefore need to take the necessary measures.

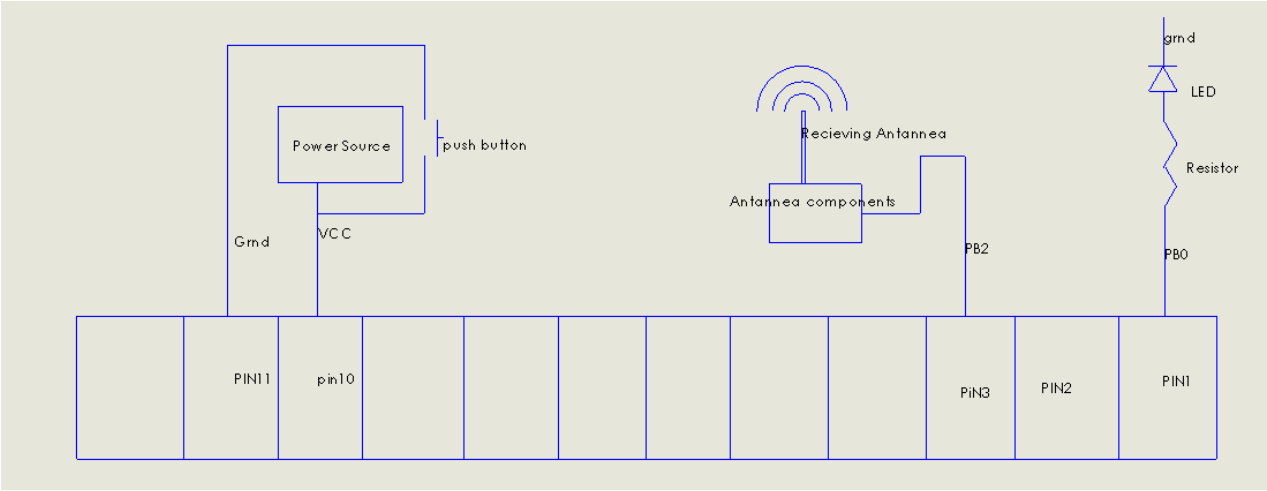

Figure 17: Pin connections of the Microcontroller

```
//Header files to be used in supporting code functions
#include <avr/io.h>//The microcontroller header file
#include <avr/interrupt.h>// header file for interrupt
#include <util/delay.h>// header file to implement delays in the code

int main(void)
{
    sei(); //enables interrupt
    DDRB |=1 << PINB0; //pin0 set as output
    while(1)//polling or endless loop
    {
    }
}
ISR(INT2_vect)// portPB2 used as external interrupt
{
        PORTB ^=1 <<PINB0;// this assignment switch the LED on and off
                //^ flips the bit after the delay from either
                //1 to 0 or vice versa
        _delay_ms(100); // this set the delay to 100 milliseconds
}
```

Figure 18: Code to program the Microcontroller

The complete receiver design will encompass both the BPSK receiver and microcontroller. The below figure 19 shows the proposed design for the RTVC package receiver. The on and off button are connected to the supply voltage, the LED is connected to the port of the microcontroller, and the antenna belongs the BPSK receiver.

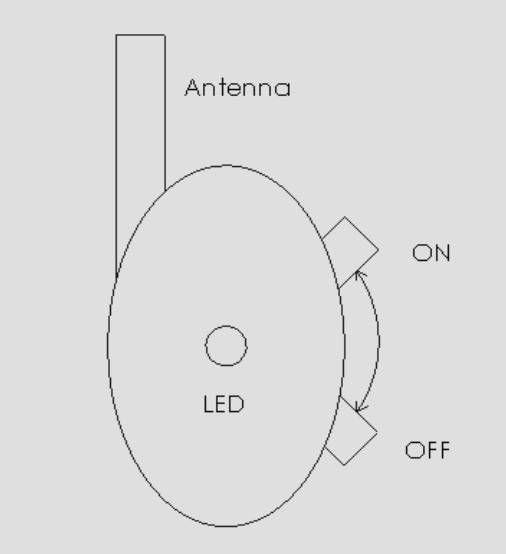

Figure 19: BPSK Package Receiver

5 Future Design Ideas

5.1 Changing the Microcontroller Code to Use Serial Ports

The RTVC package currently uses parallel ports to read information sent from the transmitter. The receiver's microcontroller parallel ports accept the transmitted signal, and cause the LED to flash using parallel interrupts. This design is used as an intermediary step, as any signal passing through the receiver will activate the LED.

Serial ports are used to store and transmit data into buffers. In order to ensure the frequency being received is RTVC's transmitted signal, RTVC can store data into an array and then check this data for a unique code.

In order to implement this change, the transmitter would send out a preprogrammed serial number of 0's and 1's. The serial code would be stored into the array *chkcode*[]. The transmitted serial number will get decoded by the BPSK receiver and enter into the serial port **RBUF*. **RBUF* is used as an input register, and stores the information into an array called *reccode*[] whenever data passes through *RBUF, by triggering the interrupt. *reccode*[] is used to receive the received signals from **RBUF*. Finally, when the array is full, the interrupt handler for **RBUF* executes a check with the stored serial number. When data matches the handler executes a second check, which then checks each bit after to see if the data is in the correct sequence. If no sequence is found, the array resets and starts storing data into its 1st position. This implementation uses pseudo code similar to section 3.4, and the implementation will vary between microcontrollers.

5.2 Two-Way Communication

BPSK transceiver used in this report has a maximum transmitting distance of 1.8 km. Considering the limited transition distance, for future implementation we propose using cellular network to allow larger transmitting distance with a two-way communication capability. In addition to expanding transmission distance, cellular communication will

provide the user the ability to react to theft warning by sending SMS, “STOP ENGINE”, from mobile device, hence making it a two-way communication. The SMS will be sent to a phone number dedicated to a device in the vehicle. The device will interpret the SMS and transmit CAN message to an engine ECU to shut-down the engine.

Based on a prototype done by Huaqun Guo, Lek Heng Ngoh, Yong Dong Wu, and Joseph Chee Ming Teo in Secure Wireless vehicles monitoring and Control, they implemented two-way communication [18]. They used Soekris Net4801 as a central controller platform, D-Link DWL-G520 wireless card and iTegno GPRS Modem for sending wireless signal to a mobile device. By basing this prototype, we can further expand our implementation of two-way communication.

6 Conclusion

This document has been a technical introduction of the Real-Time Vehicle Communication Package. The RTVC prototype will cost approximately $249.00, and can be successfully designed in four months. With this one-time cost, RTVC proves to be an inexpensive vehicle security device that will reduce vehicle theft.

7 References

[1] C. Hamacher, Z. Vranesic, S. Zaky, N. Manjikian. *Computer Organization and Embedded Systems Sixth Edition* . New York: McGraw-Hill, 1990, pp. 394, 397.

[2] Canadian Underwriter. (2010, May). Recovery rate for stolen vehicles drastically reduced, suggesting organized crime involvement: OPP. Retrieved 27 Nov 2012, from http://www.canadianunderwriter.ca/news/recovery-rate-for-stolen-vehicles-drastically-reduced-suggesting-organized-crime-involvement-opp/1000371064/

[3] S. Ashe. (2010, Sept.). A recently released FBI 2009 Crime Statistics report indicates the national vehicle theft rate is decreasing, but the rate of recovery for stolen vehicles is at an 25 year low. Retrieved 15 Nov 2012, from http://reviews.cnet.com/8301-13746_7-20016416-48.html

[4] T. Tozer. (2002, Nov. 19). Introduction to Link Budgets. Retrieved 28 Nov 2012, from http://www.satcom.co.uk/article.asp?article=21

[5] R. Swenson. (n.d). Terrestrial Link Budgets for Digital Communications. Retrieved 28 Nov 2012, from http://moodle.uvic.ca/file.php/22620/LinkBudgets-KF4DII.pdf

[6] Jarrett. (2011, April). basics: walking distance to transit. Retrieved 15 Nov 2012, from http://www.humantransit.org/2011/04/basics-walking-distance-to-transit.html

[7] D. Cassel. (2007). Amateur Radio Bands. Retrieved 27 Nov 2012, from http://www.eham.net/newham/bands

[8] (n.d). Canadian HF Band Plan. Retrieved 27 Nov 2012, from http://www.eham.net/newham/hfbandCanada.htm

[9] (n.d). Radio Specialists. Retrieved 27 Nov 2012, from http://www.radiospecialists.com.au/mobile%20antennas.htm

[10] (2012). 1" "Short Stubby" antenna. 2.4GHz, SMA. Retrieved 27 Nov 2012, from http://www.semiconductorstore.com/cart/pc/viewPrd.asp?idproduct=42988

[11] M. Dauvergne. (2008, Dec.). Motor vehicle theft in Canada, 2007. Retrieved 15 Nov 2012, from http://www.statcan.gc.ca/pub/85-002-x/2008010/article/10744-eng.htm

[12] (n.d). Signals and Modulation. Retrieved 16 Oct 2012, from http://moodle.uvic.ca/file.php/22620/IQ_review_2.pdf

[13] (n.d) CAN Protocol. Retrieved 19 Nov 2012, from http://www.can-cia.org/index.php?id=systemdesign-can-protocol

[14] Microchip Technology, Inc. (2007). PIC18F2455/2550/4455/4550 Data Sheet. Retrieved 19 Nov 2012, from http://www.microchip.com

[15] Microchip Technology Inc. (2007). dsPIC30F4011/4012 Data Sheet High performance Digital Signal Controllers. Retrieved 19 Nov 2012, from http://ww1.microchip.com/downloads/en/DeviceDoc/70135E.pdf

[16] Patrick Hood-Daniel. (n.d). Learn the fundamentals of Mechatronics. In newbieHack. Retrieved 28 Nov 2012, from http://www.newbieHack.com/default.aspx

[17] (2006) ATMEGA32-16PU - 8-bit AVR Microcontroller datasheet. Retrieved 28 Nov 2012, from http://www.alldatasheet.netpdf/pdf/171448/ATMEL/ATMEGA32

[18] H. Guo, Lngoh, Y. Wu, J.Teo. Secure Wireless Vehicle Monitoring and Control. (2009) Retrieved 20 Nov 2012, from http://ieeexplore.ieee.org.ezproxy.library.uvic.ca/xpls/abs_all.jsp?arnumber=5394135&tag=1

Log book grade (25%): _____

Report grade (75%): _____

Total Grade (100%): _____

Supervisor's Comments:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____ _____ _____

Supervisor's name (Print) Signature Date

Notes for the supervisor:

1. Please return the marked hard copy to Prof. Tao Lu by Monday, December 10.
2. Attached additional pages for comments if necessary.