Application Note AN2308

Remote Keyless Entry Car Alarm with Floating Code

Author: Volodymyr Sokil
Associated Project: Yes
Associated Part Family: CYWUSB6953, CY8C21x34, CY8C29x66
Software Version: PSoC Designer 4.2
Associated Application Notes: AN2268, AN2307

Abstract

This Application Note is an example of a remote keyless entry car alarm system. This system is based on a strong cryptography authentication method and uses a bidirectional Cypress WirelessUSB™ link. It can be used as the base for a complete car alarm or a secure remote keyless entry system (door opener, access control, etc).

Introduction

With the worldwide increase in the number of cars, there has been a corresponding increase in the number of car thefts. Car security systems can prevent or deter these crimes. A car security system can consist of different devices and subsystems, such as immobilizers, remote keyless entry, alarms, and tracking subsystems.

Remote keyless entry systems are also used for other purposes. Garage door control, home security (including room access control), and secure remote controls for devices are just some of the uses of remote keyless entry systems. These systems have different functionality, but use the same type of security components as the car entry system described in this Application Note.

Theoretically, any antitheft system can be broken. Otherwise, if you lost the key you could not disable the system and get a replacement key; you would have to replace the car. Immobilizers and GPS tracking systems can be defeated, for example, by placing the car inside a shielded trailer. However, loading a car into a trailer takes time. The alarm system reduces the amount of time that a criminal has to work before being detected. Criminals must know how to swiftly disable the alarm system or use a different method.

Car alarm systems usually consist of two parts; a sensor network and a remote keyless entry system. If the sensor network is correctly designed and mounted so that a criminal does not have physical access to the sensors or the alarm, the keyless entry system becomes the weakest part of the system and the focus of efforts to defeat the system.

The remote keyless entry provides security on two levels; the low level and the high level. The low level is the radio signal transmission. The high level is the secure data transmission channel. The most important part of the secure data transmission channel is the remote control authentication protocol.

Radio Channel Security

Most modern car alarm systems use the radio channel with band pass modulation, such as amplitude shift keying (ASK), frequency shift keying (FSK), phase shift keying (PSK), or a combination of these. The radio channel uses a narrow-band radio signal spectrum. Using special grabbers these signals can be easily intercepted and stored by criminals. If the car alarm system uses a one-way communication link with static code, the criminal can simply replay the stored signal at a later time to gain access to the car.

If the alarm system uses two-way communication and floating code, but does not use reply protection, the criminal can simply transmit all possible reply codes over a period of time to eventually gain access to the car.

Finally, even if the criminals cannot directly use intercepted signals, they can analyze the authentication protocol with the help of the intercepted signals, other obtained data, and powerful computer networks, to eventually learn how to defeat the system.

Therefore, none of these methods using narrow-band radio provides sufficient protection.

The proposed solution to this problem is to use a spread spectrum radio channel. The spread spectrum radio channel system must satisfy three conditions:

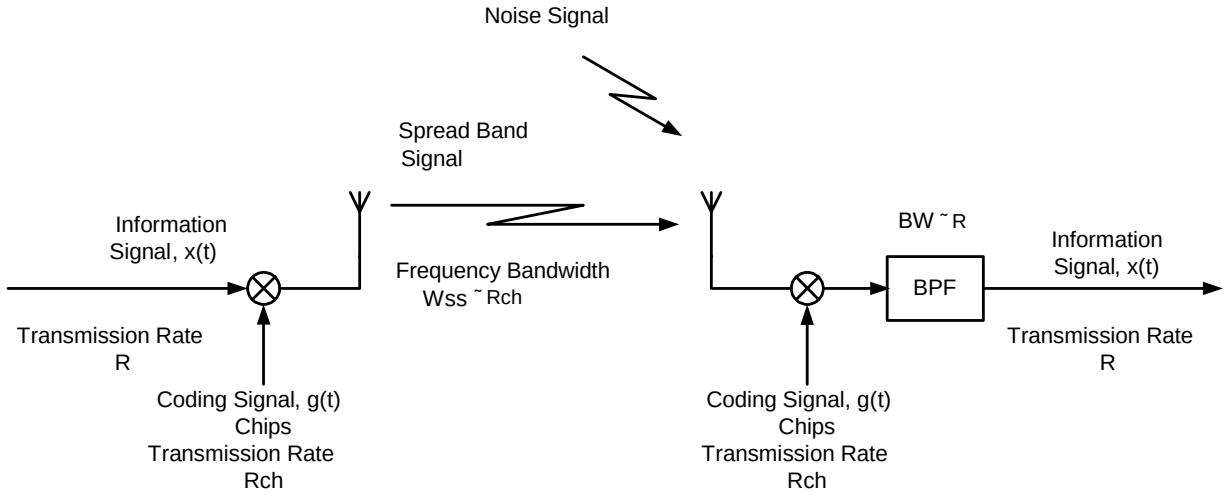
- The frequency band must be wider than necessary for data transmission.
- The spectrum spreading must use a coding signal that is not dependant on the data transmitted.
- The receiver must reconstruct the signal using the receiving signal and a synchronized copy of the coding signal.

There are three methods of spectrum spreading; direct sequencing (DS), frequency hopping (FH), and time hopping (TH). A combination of these methods can be used.

This project uses Cypress WirelessUSB™ radio modems, based on the CYWUSB6934 Radio SoC. This is a highly integrated, low cost radio transceiver, which operates in the unlicensed Industrial, Scientific and Medical (ISM) band (2.4–2.483 GHz). These modems can provide radio link ranges up to 10 meters without an external power amplifier and up to 1 km when an external power amplifier is used. More detailed information on these radio modems can be found in the CYWUSB6932/CYWUSB6934 data sheets.

As an alternative to the separate radio modems, the new Cypress Semiconductor CYWUSB6953 WirelessUSB™ PRoC™ (Programmable Radio System-on-Chip) can be used. The CYWUSB6953 WirelessUSB PRoC is the world's first low cost Flash programmable microcontroller with an integrated ISM band radio transceiver. More detailed information on this unique IC can be found in the data sheet. The CYWUSB6953 is the ideal solution for this system.

Figure 1. Direct Sequence Spread Spectrum Flowchart



These modems use the Direct Sequence Spread Spectrum (DSSS) radio channel. A flowchart of this method is shown in Figure 1.

Information signal x(t) is modulated by multiplying it with the coding signal g(t). Two functions multiplied in the time domain corresponds to its convolution in the frequency domain.

$$x(t)\cdot g(t) \leftrightarrow X(\omega) * G(\omega) \quad \textbf{(1)}$$

Since the information signal is narrow-band (in comparison to the coding signal), the result of multiplying x(t)·g(t) is approximately equal to the coding signal bandwidth. The result of this is that the information signal is distributed in the defined spectral region more evenly and with less density than it is with band pass modulation.

An additional noise signal can be added to mask the useful signal. This signal can be generated by a second spread-spectrum signal with a different coding sequence.

The demodulator multiplies the receiver signal with a synchronized copy of the coding signal and obtains the restricted signal. A band pass filter (BPF) is used to remove stray signals.

Single multiplication with g(t) results in spectrum spreading; double multiplication results in spectrum restriction. Because the noise signal is multiplied only once, the system becomes simultaneously more resistant to noise and interception. Criminals who do not have access to a synchronized copy of coding signal will find the spread spectrum information signal is lost in the noise.

The WUSB modems use a random bit sequence as the coding signal. The coding signal is stored in both the transmitter and receiver. The length of this sequence depends on data bit rate. For example, if the data bit rate is 15,625 kbps, the coding signal length is 64 bits. Because there are 2^{63} variants that a criminal would have to try, stealing the car with a brute force attack would require large amounts of computing power and time.

There are other methods the criminal may use to obtain the coding signal. These methods are too complex to explain in this Application Note. Nevertheless, if the criminal obtains the coding signal, they still have not defeated the security system. There is a second layer of authentication to prevent these kinds of attacks.

Simple Lightweight Authentication Protocol

If criminals manage to defeat the security provided by the radio channel, they can try to simply reuse the intercepted signals or develop a remote control emulator. For protection against this type of attack, you employ an authentication protocol. The authentication protocol provides a secure channel between the car and an authorized remote control.

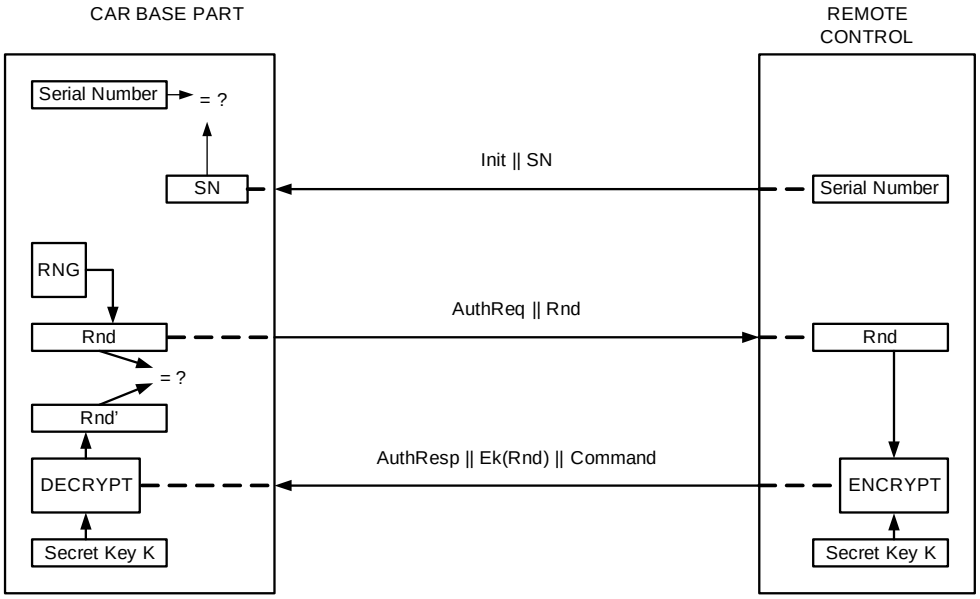
Authentication is the process of determining whether something is what it says it is. Usually the subject of authentication confirms its identity by demonstrating knowledge of some secret information such as a key or password.

One of the most popular and well known methods is simple authentication based on a nonexpendable password. As described previously, this password can be intercepted and reused. An authentication procedure based on the one-time passwords is more effective. For each new transaction, a new password is used. Because one-time passwords are only valid for a single transaction, intercepting the password is useless. It cannot be reused.

One-time password authentication is based on the challenge-handshake protocol. To check the authenticity of the response, the authenticator (A) sends a challenge message to the peer (B). The challenge consists of some unpredictable value, x (a random number, for example). B responds with a value calculated using some function, f(x), known to both A and B. The authenticator checks the response against its own calculation of the expected function value. If the values match, the authenticator can be sure of the identity of the peer, B.

The system detailed in this Application Note uses a modified challenge-handshake protocol. Since the goal of this Application Note is a low-cost bidirectional car alarm system with feedback, the protocol provides two operational modes; normal operation and alarm operation. The flowcharts of these two operational modes are shown in Figure 2 and Figure 3.

Figure 2. Authentication Protocol – Normal Operation Mode



In normal operation mode shown in Figure 2, the transaction is initiated by the remote control (RC). This mode is used during normal operation such as locking or unlocking the car by remote control.

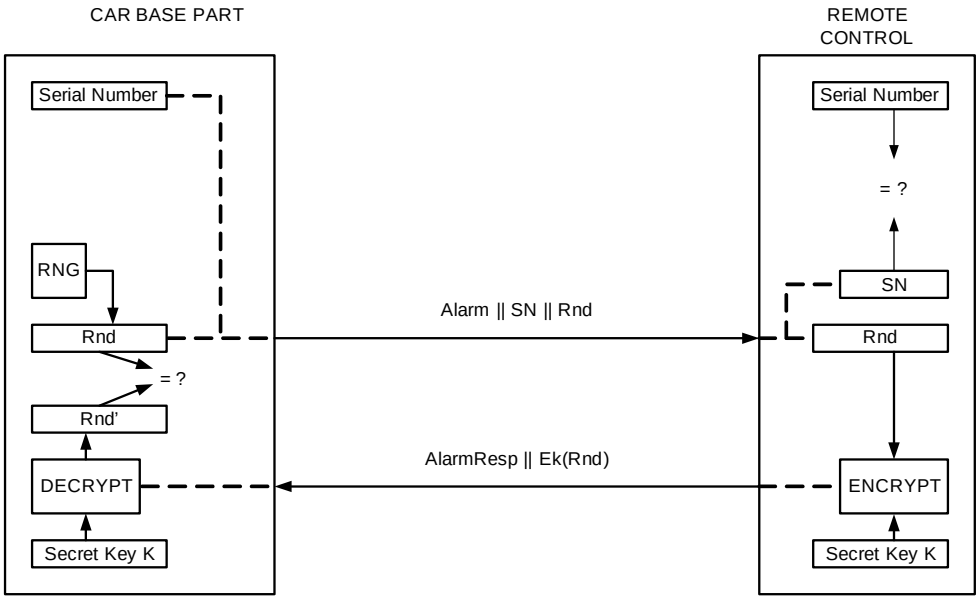
The protocol's first step in normal operation mode is to identify the remote control. The RC sends its serial number as an identifier.

If the base recognizes the RC serial number, it generates a random number and sends the authentication request. In this protocol, instead of a function, the remote control encrypts the random number. Encryption is the algorithmic process of obscuring data so there is a low probability of being able to use it without a confidential process or key. The RC encrypts the received random number and sends it back to the car base station along with a command ("unlock the doors" for example).

The base station extracts the encrypted random number from the authentication response and decrypts it. If the decrypted and stored values are not equal to each other, the command is ignored.

In alarm operation mode shown in Figure 3, the transaction is initiated by the car base station. This mode is used if the sensor network detects a break-in attempt or other alarm event. In this case the alarm signal must be transmitted to the remote control. Because the transaction is initiated by the car base station, the RC must identify the base station by the same serial number procedure used previously. The authentication mechanism is the same as in normal operation mode.

Figure 3. Authentication Protocol – Alarm Operation Mode



Authentication Protocol

The authentication protocol uses two cryptography building blocks; a random number generator and symmetric encryption. This system can use either a true random number generator or a pseudo-random number generator as long as it has a sufficiently long generation period. More detailed information about random and pseudo-random numbers and an example of true random number generator implementation can be found in AN2307, “Hardware Random Number Generator.”

The encryption algorithm should have the following characteristics:

- It must be resistant to known cryptanalysis attacks (such as brute force attack, linear and differential cryptanalysis).
- It must be compact so that it does not consume more memory than is available on the chosen Cypress PSoC device.
- It should be relatively fast, though this requirement is not absolutely necessary.

The chosen algorithm for this application is the symmetric block encryption algorithm, RC5. It was designed by Ron Rivest in 1994. Symmetric means the encryption and decryption of the data is done with the same secret key. A block encryption algorithm processes a block of plain data as a whole. RC5 is a parameterized algorithm with a variable block size, a variable number of transformation rounds, and a variable key size. The block size depends on word’s length. Allowable choices for the block size are 32 bits, 64 bits, and 128 bits (corresponding to machine word lengths of 16, 32, and 64 bits). The number of rounds can range from 0 to 255. The key can range from 0 bits to 2040 bits in size.

Such built-in variability provides efficiency and flexibility at all levels of security. RC5 algorithms are designated RC5-w/r/k, where:

- w is the machine word length in bits,
- r is the number of rounds,
- b is the key size in bytes.

For example, RC5-32/12/5 uses 32-bit words (64-bit block length), 12 rounds, and a 40-bit (5 byte) key. This is what is used in this application.

There are three routines in RC5:

- Key expansion
- Encryption
- Decryption

In the key-expansion routine, the secret key is expanded to fill a key table. The size of this table depends on the number of rounds. The encryption and decryption routines use the same key table. The key-expansion routine flowchart is shown in Figure 4. During this process, *t* sub-keys are calculated as the result of sufficiently complicated operations with a secret key. Each round uses two sub-keys; another two sub-keys are used for an additional operation, which is not a part of any round. So the number of sub-keys is t = 2 · r + 2. The length of each sub-key is equal to *w* bits.

Sub-keys are stored in the array of *t* words. Elements of this array are designated as:

$$S[0], S[1], \ldots, S[t-1]$$

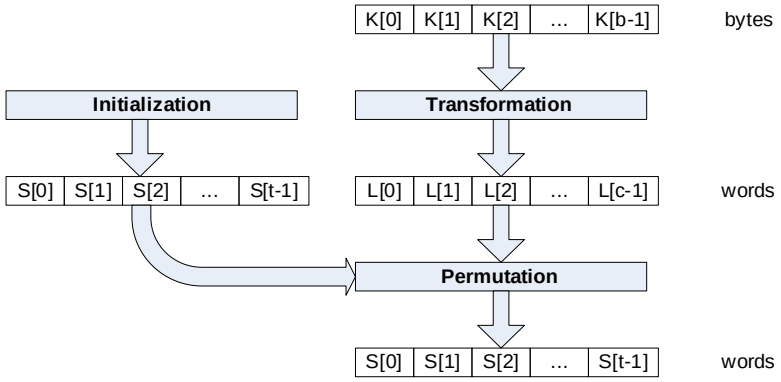
Using the parameters *r* and *w* as the input data, array *s* is initialized by predefined pseudo-random values. After that the secret key *K* is transformed in the array of words *L*.

$$K[0..b-1]$$

$$L[0..c-1]$$

If *b* is not divisible by *w*, the right part of *L* will stay equal to zero. Finally, the permutation operation with elements of the two arrays *L* and *S* is executed. The result is that finished values in array *S* are obtained.

Figure 4. RC5 Key-Expansion Routine Flowchart



The encryption routine consists of three primitive operations:

- Integer addition
- Bitwise XOR
- Variable left rotation

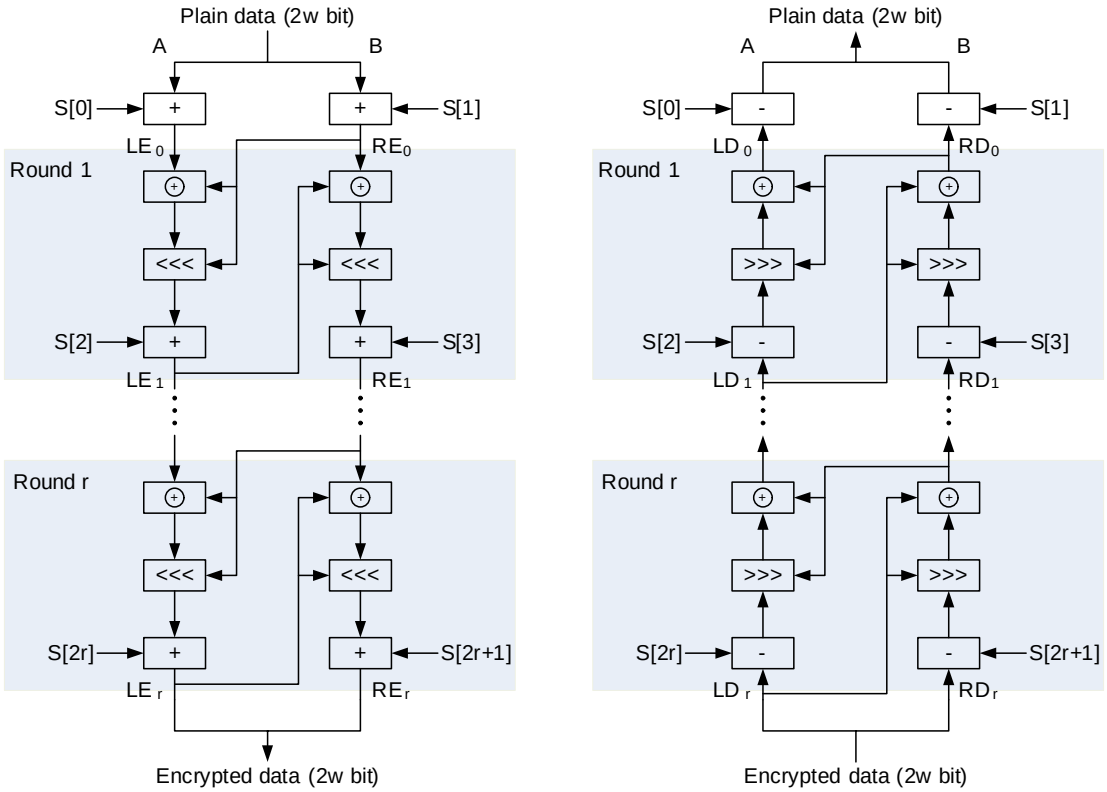
The decryption routine consists of three inverse primitive operations:

- Integer subtraction
- Bitwise XOR
- Variable right rotation

The encryption routine flowchart is shown on the left side of Figure 5. The structure of this algorithm is different from the classical Feistel network structure. At the start, plain data is stored in two registers A and B with *w* bits length. Each of *r* rounds consists of a substitution, a permutation, and another substitution, which depends on the key. Note that during each round, both data blocks' halves are changed.

The decryption routine shown on the right side of Figure 5 is the reverse of the encryption routine. More detailed information about the RC5 encryption algorithm can be found in the Internet Engineering Task Force (IETF) document RFC 2040, *The RC5, RC5-CBC, RC5-CBC-Pad, and RC5-CTS Algorithms*.

Figure 5. RC5 Encryption And Decryption Routine Flowcharts



Prototype System Hardware Implementation

To check the performance of the proposed solution, a very simple prototype system was developed. This system consists of two boards; car base station and RC. The board's schematic circuits are shown in Figure 6 and Figure 7.

RC Schematic

The RC prototype board is very simple, but sufficient for testing. It contains only a PSoC® CY8C21534 (U4) device, a radio modem connector, a power supply circuit, and a user interface.

The WirelessUSB radio modems are separate, ready-to-use radio modems. The JUNO-L WirelessUSB™ radio modules are produced by Unigen Corporation (www.unigen.com), a Cypress partner. The radio module board is connected to the RC board via the J3 connector.

The user interface is represented by two buttons, SW2 and SW3 (for different operations), and LED D7, intended as an indicator that an alarm signal was received from the car base station. The low-drop linear regulator U5 provides a 3.3V power supply for the all devices' components.

Note The PSoC on-board switch mode pump (SMP) can be used if the battery power is 3.0V. It prolongs battery life.

Figure 6. RC Schematic

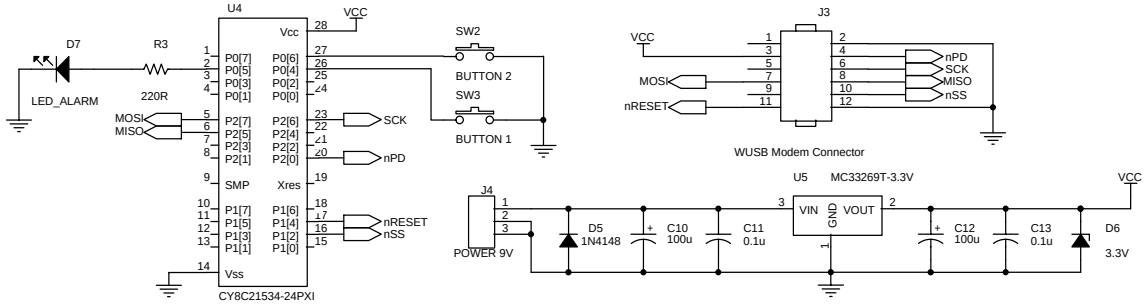
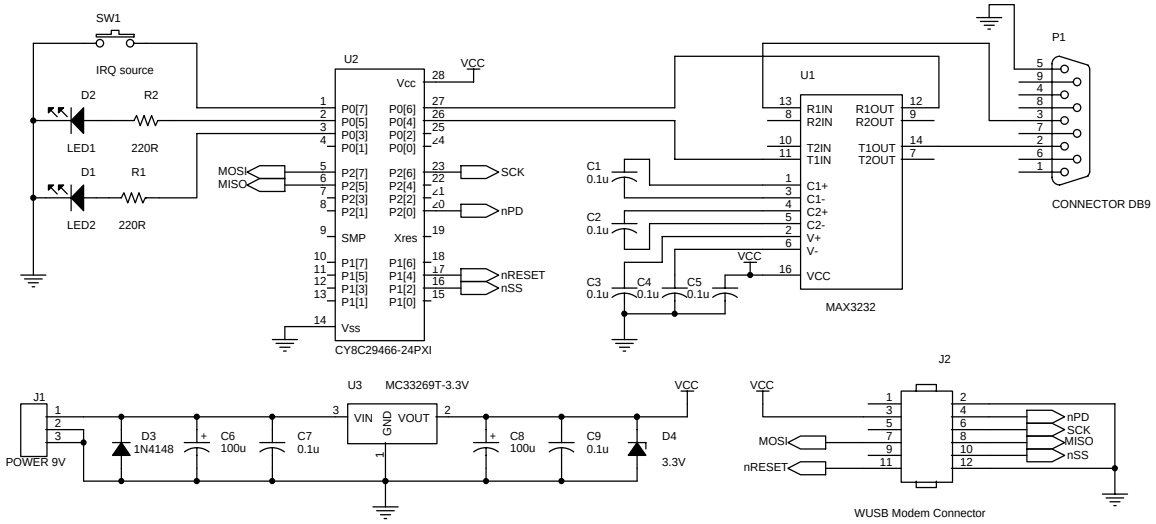


Figure 7. Car Base Station Schematic



Car Base Station Schematic

On the car base station prototype board, a level translator, U1, is added so that the device can be connected to a PC via an RS232 cable connected to the PC's COM port. This interface is used for debugging purposes only and is not required in the end design. The base station requires more hardware and memory than the remote control, so the PSoC CY8C29466 is used.

LEDs D1-D2 indicate when switches SW2-SW3 are pressed on the RC. Switch SW1 is used to simulate an alarm signal event.

The radio module board is connected to the base station board via the J2 connector.

Prototype System PSoC Device Internals

The internal structure in the test configuration in PSoC Designer™ is also very simple. The remote control user module placement and interconnectivity is shown in Figure 8. This structure consists of one user module, SPIM, and a SleepTimer.

The SPIM, which is placed in DCB02, provides communication with the radio modem. The SPI link baud rate is 1.5 Mbps.

The SleepTimer period is set to 0.125s. It provides the timeout during data receiving process.

Note Ideally, this design would be ported to the new Cypress Semiconductor PRoC (Programmable Radio-on-a-Chip™) device, CYWUSB6953.

Figure 8. RC PSoC Internal User Module Configuration

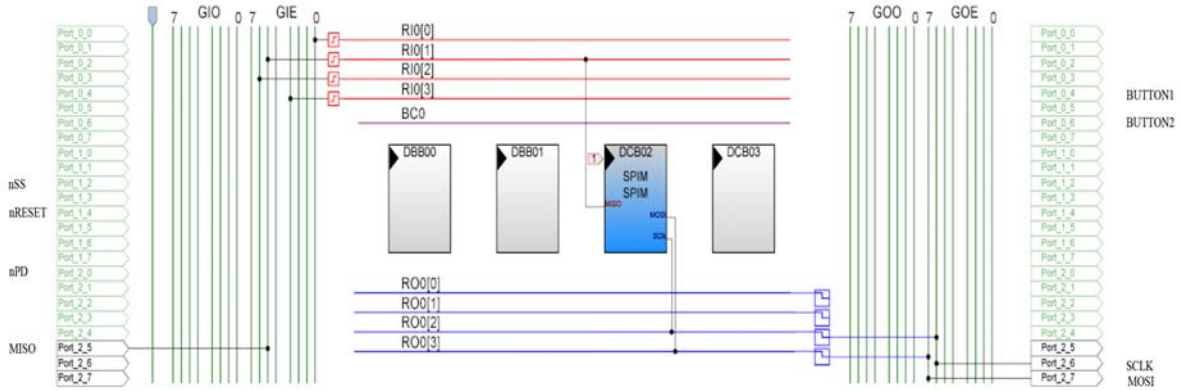
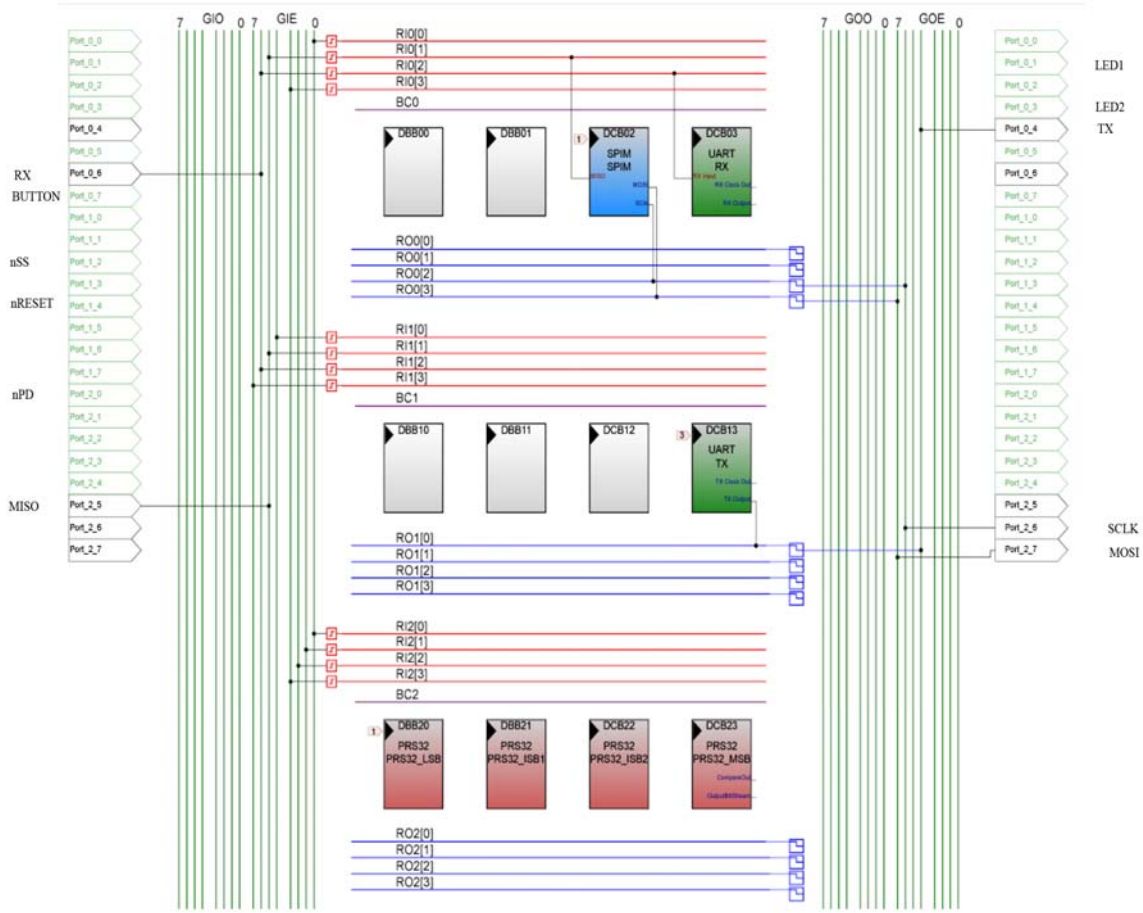


Figure 9. Car Base Station PSoC Internal User Module Configuration



The car base station internal structure consists of three user modules, SPIM, UART, PRS32, and the SleepTimer. The SPIM and SleepTimer have the same function, placement, and parameters as in the RC internal structure. The UART provides the communication with the PC. The baud rate of the PC link is 115,200 bps.

The random number generator uses the PRS32 User Module, which is placed in DBB20, DBB21, DCB22, and DCB23. This generator produces a random sequence with a period of 2^{32}-1.

Prototype System Firmware

The prototype system firmware implements secure radio channel support and general device control. It consists of several modules that serve distinct functions such as data exchange via the radio link between the RC and the car base station, authentication protocol, and user interface support.

The data exchange module consists of:

- Low-level subroutines that provide communication with radio modem through the SPI interface
- Subroutines that implement error correction, coding, and decoding algorithms
- A serial transceiver

The first and second parts are described in detail in the AN2268, “Forward Error Correction using a WirelessUSB Radio System-on-Chip (SoC) Modem.”

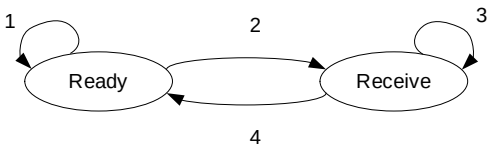
The serial receiver control unit state diagram is shown in Figure 10.

Initially, the receiver is in READY state. The receiver stays at this state until a data packet start byte is received (1). After the start byte is received, the receiver moves to the RECEIVING state (2). This state will be preserved until the packet’s stop byte is received (3). When this happens, the receiver returns to READY state (4).

The receiver also can return to the READY state (4) if any of the following events occur:

- A byte is received with an error.
- More than the buffer-size packet bytes are received without a stop byte.
- The timeout expires.

Figure 10. Receiver’s Control Unit State Diagram



The transmitter is very simple. At the start of transmission, it sends one preamble byte (0x55) and a start byte (0xAC). After the data packet transmission, a stop byte (0x53) is sent.

The command set shown in Table 1, is used for the authentication protocol. Each command has fixed structure, shown in Figure 11. The shaded fields are mandatory for all commands. Other fields are used by the commands as shown in Table 1.

Table 1. Authentication Protocol Command Set

OpCode	Command	Fields	Description
1	Init_Command	OP, SN	Communication start command.
2	AuthReq_Command	OP, SN, PAS	Sent by the authenticator to the peer as the authentication request.
3	AuthResp_Command	OP, SN, PAS	Sent by the peer as the authentication response to the authenticator.
4	Finish_Command	OP, SN, INF	Sent by the authenticator to indicate successful authentication. Can contain additional information for the peer (for example, car status information for the RC).
5	Sync_Command	OP, SN	Resynchronization command.
6	Alarm_Command	OP, SN, PAS	Sent by the base station to signal an alarm.
7	AlarmResp_Command	OP, SN, PAS	Sent by the RC to acknowledge alarm receipt.

Figure 11. General Command Format

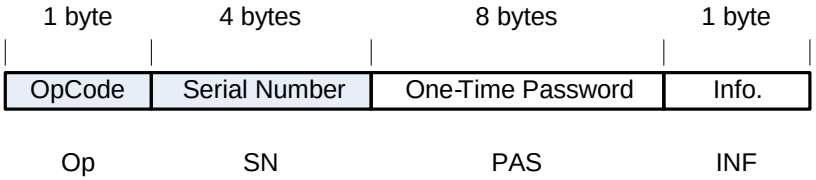
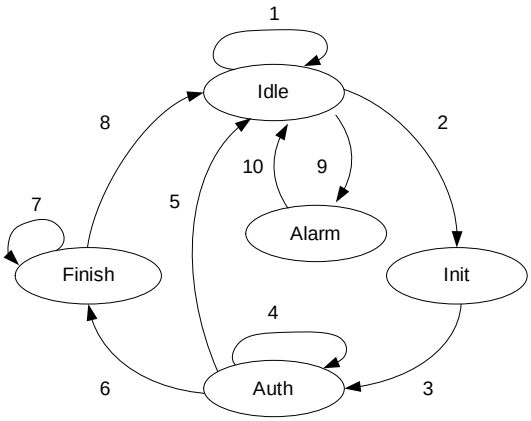


Figure 12. RC's Control Unit State Diagram



The behavior of the remote control is defined by the RC control unit, shown in Figure 12. The initial state of the RC's control unit is the IDLE state. The RC remains in IDLE state (1) until any of the following events fire:

- One of two buttons is pressed. This is normal operation mode.
- The alarm command from the car base station is received. This is alarm operation mode.

If the user presses a button (for example, "unlock car") the control unit moves to the INIT state (2). In this state the Init_Command is formed and sent to the base station. The control unit goes to the AUTH state (3).

It stays at AUTH state (4) until the AuthReq_Command is received or the timeout expires. If the timeout expires, the control unit goes back to the IDLE state (5). When the AuthReq_command is received, the password from the AuthReq_Command is encrypted and the results are sent with the AuthResp_Command to the base station. The control unit moves to the last state, FINISH (6).

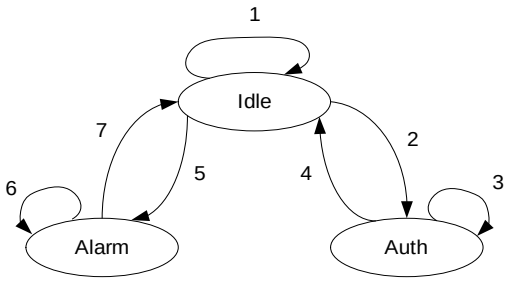
It stays at FINISH state (7) until the Finish_Command is received or timeout is elapsed. After that the control unit returns to the IDLE state (8).

In the alarm operation mode, the control unit moves to the ALARM state (9). The RC transmits the Alarm_Resp command to the car base station and goes back to the IDLE state (10).

The car base station control unit is shown in Figure 13. The initial state of the base station control unit is the IDLE state. In normal operation mode the base station receives the Init_Command from RC and goes to the AUTH state (2). The AuthReq_Command is formed and sent to the RC.

The control unit stays at this state (3) until the AuthResp_Command is received or the timeout expires. If the timeout expires, the base station goes back to the IDLE state (4). When the AuthResp_Command is received, the authentication check is executed. If the authentication passes, the RC command is executed. After that, the Finish_Command is formed and sent to the RC. The control unit returns to the IDLE state (4).

Figure 13. Base Station's Control Unit State Diagram



If the alarm interrupt signal from the sensor's network is detected, the control unit moves to the ALARM state (5). In this state, the Alarm_command is periodically sent to the RC (the period is 1s). This state will be preserved (6) until the AlarmResp_Command from RC is received. After that, the control unit returns to the IDLE state (7).

Conclusion

This Application Note describes the communication module for a car alarm system with a bidirectional interface. This module provides security on two levels. The first level is the interception-proof, noise-immune radio channel using the spread spectrum radio IC transceiver. The second level is the strong authentication for all RC commands. The authentication protocol is based on one-time passwords and a challenge-handshake procedure.

The authentication protocol implementation uses the RC5 encryption algorithm with a 40-bit key to satisfy US export restrictions. But key length can be easily increased up to 128 bits. Moreover, the whole algorithm can be easily replaced by any other customer-designed encryption algorithm with a 64-bit block size.

The functionality of the communication module can easily be modified and expanded. For example, it can be adapted to work with several RCs. To do this you would add a table with authorized RC serial numbers and corresponding secret keys to the firmware of the car base station. The general system behavior can be changed so the system can be used in other keyless entry systems.

Note The remote control application should contain a power management system to extend battery life. A sleep mode with an external key wake-up can be used for sending commands from the RC to the base station. A sleep timer wake-up can be used so that the RC periodically checks for alarm messages from the base station. The PSoC device SMP allows prolonging the battery life by boosting the voltage from a moderately discharged cell. The RC can be powered by one low-cost AAA alkaline, the SMP boots voltage from 1.3..1.6V to a nominal 3.3V. To boost the SMP current for the WUSB device, an external MOSFET transistor can be used. This allows you to integrate the internal step-up converter to save money. The PSoC internal comparator can be used to check for low battery. This allows you to create a single package solution with rich features at an affordable price.

About the Author

Name: Volodymyr Sokil

Title: Post-Graduate Student

Background: Volodymyr earned a computer-engineering diploma in 2001 from National University "Lvivska Polytechnika" (Lviv, Ukraine), and is prolonging his study as a post-graduate student. His interests include embedded systems design and information security.

Contact: sokilvm@ukr.net

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com