Chapter 1.
Background

1.1 Introduction

This project is intended to demonstrate a concept from a New Zealand company called the Gallagher Group. The Group is committed to the research, design, building and selling of modern farm management and security technologies. In order to be more competitive and remain leaders in the electric fence market, the Gallagher Group has identified the need to enhance their present tools such as farm monitoring devices (which constitute part of a farm management system). Their vision is to achieve centralized control of different devices attached to electric fences from a human interface that can be accessed in the farmer's home. For this purpose it is desirable for the farmer's personal computer (PC) to be such an interface to enable communication with the different devices attached to the electric fence.

Industry experience shows us the path to follow regarding choice of a communication medium. Previous research and technology presently available provides the first step to implementing a prototype that enables communication through a serial link. The next step is to use a wireless link. In the following chapters, different modules that were considered to undertake this project will be presented, including the problems faced, project specifications and project limitations.

Firstly, the modules that make up parts of an electric fence system will be introduced.

1.2 The Electric Fence

Electric fences constitute a psychological barrier for animal containment. A pulsed electric current is sent along the fence wire, at a rate of about one pulse per second. The animal receives a shock when it completes the circuit between the fence and the ground (Figure 1.2.1) [1].

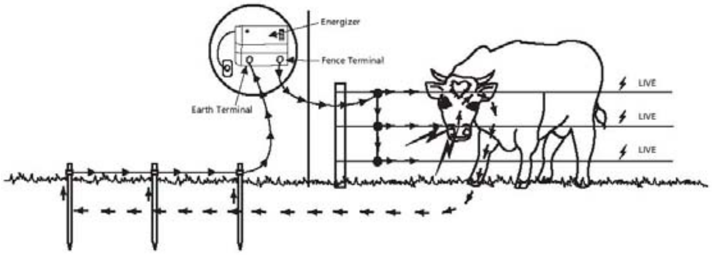


Figure 1.2.1: The Animal Touches the Fence and Receives a Shock

In Figure 1.2.2 [2] the basic components that constitute an electric fence are illustrated and the following section will explain them in more detail.

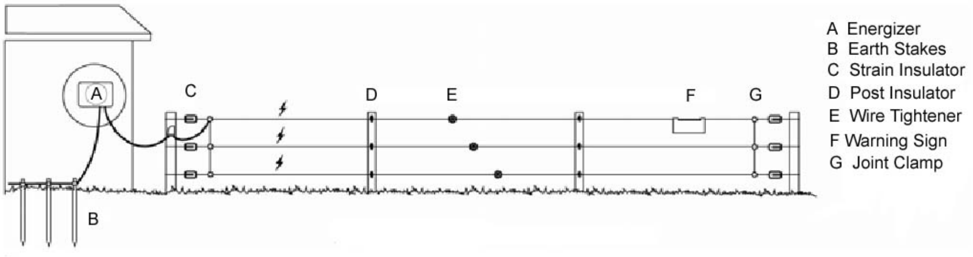


Figure 1.2.2: Basic Elements of a Permanent Electric Fence

[1] Gallagher Group LTD (2005). *How to?, How does an electric fence work?* Retrieved March 6, 2006 from: http://www.gallagher.co.nz/pf.downloads.aspx

[2] Gallagher Group LTD (2005). *Gallagher Power Fence System Manual* (12th ed.). Hamilton: Gallagher Group.

A) The Energizer

The energiser is the heart of the electric fence. This component supplies the electric pulses needed to energize the fence and it is also able (like some other devices) to send and receive data using the electric fence as a communication medium. Nearly every second, a short (milliseconds) kilovolt pulse is produced and sent through the fence. Emitting these pulses is the main goal of the electric fence, however, they can sometimes make the reception of data packets transmitted along the fence difficult. This is why all devices attached to the fence must be designed to prevent interference caused by pulses and to support their high voltages.

Energizers can be of different types and have different characteristics[3] according to their requirements. Some of the types of energizers are:

- Mains powered energizers
- Mains and battery powered energizers (where mains power is unreliable)
- Permanent solar powered battery energizers (where there is no mains power).
- Portable solar powered energizers (have an in-built solar panel and provide maintenance-free solar power operation)
- Portable battery powered energizers (for short term animal control).

[3] Gallagher Group LTD (2005). *Energizer's Overview.* Retrieved March 6, 2006 from: http://www.gallagher.co.nz/energizer.overview.aspx

Despite all having the same purpose, these energizers are dedicated to different applications. Their different technological characteristics enable them to accomplish the goal they were designed for in diverse ways.

B) Earth Stakes

Earth stakes (Figure 1.2.3) are metal stakes driven into ground, used to connect one pole of the electric fence to the ground and ensure the animal can complete the electric circuit by touching the ground and the fence (Figure 1.2.1). Dry or frozen soil can have poor conductivity. In these cases, earth stakes are connected directly to a wire parallel to the energized wire so that by touching both wires the animal closes the circuit and receives a shock.

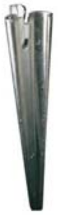


Figure 1.2.3: Earth Stake

C) Strain insulators

Strain insulators (Figure 1.2.4) are made of porcelain or plastic and are used to provide insulation on end and corner posts.

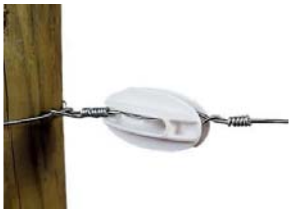


Figure 1.2.4: Strain Insulators

D) Post insulators

Although the fence is expected to be only a psychological barrier for animal containment, posts are designed to absorb pressure from animals, wind and snow. Post insulators (Figure 1.2.5) are used to ensure that wires will not detach from posts even in difficult farming environments.

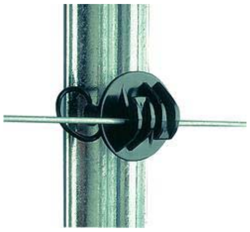


Figure 1.2.5: Post Insulator

E) Wire tighteners

Used to tighten the wire on an electric fence. (Figure 1.2.6)

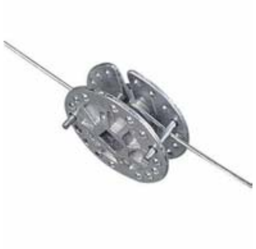


Figure 1.2.6: Wire Tightener

F) Warning Signs

These signs are visual warnings to deter people from touching the electric fence.

G) Joint Clamps

Joint clamps provide good electrical connections in order to ensure proper conductivity along the fence.

In addition to the main electric fence components, there are other devices that can be attached to the fence and make up part of the electric fence system. These can be:

- Energizers
- Remote controls to turn energizers on and off and read currents and voltages.
- Security devices (alarms).
- Control devices (gates).
- Data collection devices for environmental evaluation (e.g. grass meters).

Some of these devices have monitoring function that will be reviewed in more detail in the following section.

1.3 Devices Used to Monitor an Electric Fence

Monitoring an electric fence can be done in different ways depending on the farmer's needs. Visual indicators, strobe lamps, sirens (such as those seen in Figure 1.3.1)[4], troubleshooting or measurement devices are used for identifying problems and maintaining electric fences.

[4] Gallagher Group LTD (2005). *Gallagher Power Fence System Manual* (12th ed.). Hamilton: Gallaguer Group.

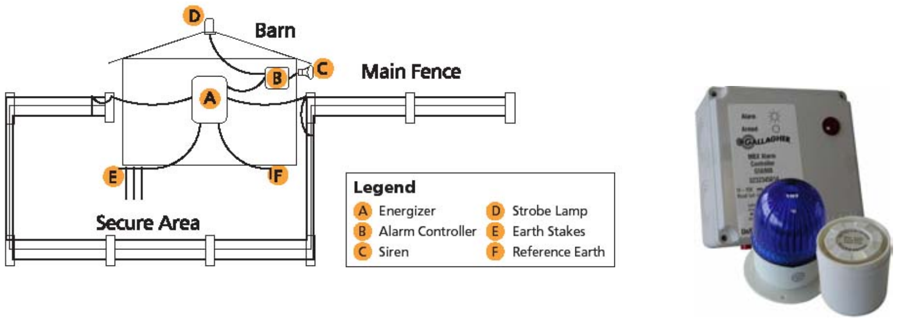


Figure 1.3.1: Monitoring Elements of an Electric Fence

- Visual Indicators shown in Figure 1.3.2 [5] and Figure 1.3.3 constitute the most basic voltage monitors of an electric fence.

Figure 1.3.2: Stafix Fence Alert

Figure 1.3.3: Gallagher's Voltage Indicator

- Traditional measurement devices such as multimeters and oscilloscopes are general tools to diagnose any electric or electronic circuit. However, they are often too general and demand users to know exactly what they are looking for when diagnosing problems on an electric fence. These tools can be useful when no dedicated tools such as log memories retrievable from energizers (which record fault events) or remote control faultfinders are available.

[5] Stafix Fence Alert (2005), *Fence Monitors.* Retrieved March 6, 2006 from: http://www.true-test.com/pel_new2

- Remote controls (Figure 1.3.4) are also used to monitor electric fences. In the context of electric fences, “remote controls” are so named because of their ability to turn an energizer on and off, or to request information from it even if it is located in a distant place. They are able to undertake these tasks by using the electric fence as a medium to transmit information. Although different remote controls can look similar and their purposes might be alike, they can have very different technical capabilities depending on their design.

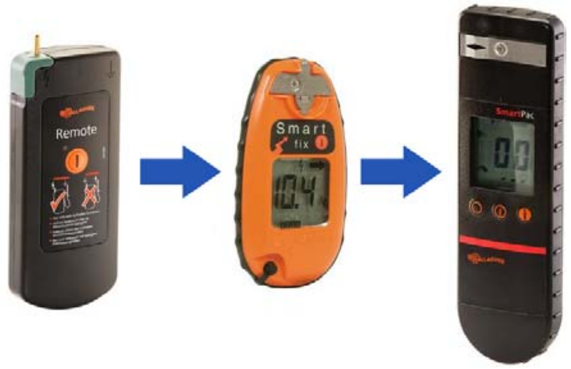


Figure 1.3.4: Gallagher’s Remote Controls Evolution

- LAN Networks (Figure 1.3.5)[6] are used to monitor electric fences in Australian zoos.

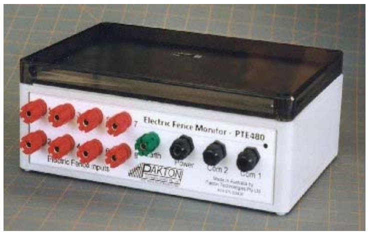


Figure 1.3.5: Pakton Fence Line Monitor

[6] Pakton (2005). *PTE0480 Electric Fence Monitor.* Retrieved March 6, 2006 from: http://www.pakton.com.au/pte0480.html

The fact that the electric fence can be used as a communication medium to transmit and receive data, creates enormous possibilities to manage any device attached to it.

1.4 Project Specifications

There are two general constraints that are considered in this thesis:

- The specific technical requirements (as in 1.4.1 Technical Requirements)
- Previous research and products (as in 1.4.2 Previous Research and Products)

1.4.1 Technical Requirements

This project's goal is to undertake a "proof-of-concept" of a wireless communication application. For this purpose it is necessary to develop a prototype able to interface devices attached to an electric fence with a PC (Figure 1.4.1).

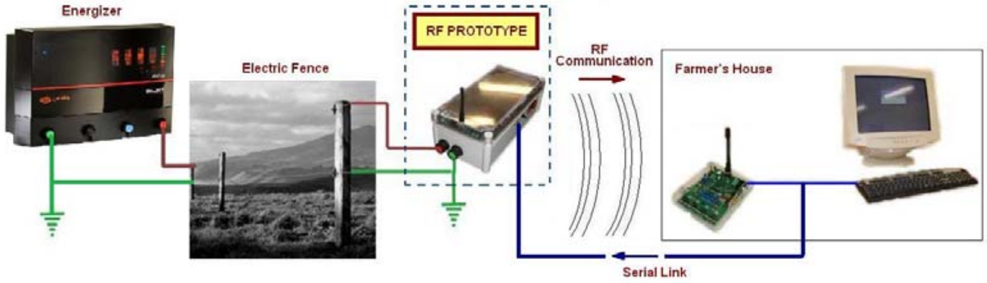


Figure 1.4.1: Concept of a Wireless Interface

The constraints are as follows:

- From the farmer’s PC it should be possible to: A) Receive data packets that travel along the electric fence and B) Transmit data packets to the electric fence.
- Frequencies for the wireless device should be programmable or configurable, considering the legal restrictions concerning the use of frequency spectra in different countries.
- Desired wireless communication range should be around 500 m.
- Existing functional products such as Gallagher’s energizers or remote-controlled electronic boards should be considered as a starting platform to develop the prototype.
- The radio frequency device must be able to transmit and receive information through walls and other obstacles.
- For the radio frequency device design technologies such as: Nordic Semiconductor www.nvlsi.no , ATMEL Smart RF www.atmel.com or Chipcon Smart RF www.chipcon.com should be considered.

1.4.2 Previous Research and Products

Previous research and market assessment by Gallagher Electronics has led to the development of new products, such as energizers and remote controls. These products have proven performance and functionality and have become mature marketable products.

From this perspective two Gallagher products are used to fulfil part of the requirements of this research: The MX7500 energizer’s logic board and the SMART PAC remote control electronic board. The adaptation of these products will be explained in the following chapters.

1.5 Expected Results

The expected results of this thesis are:

- To make a prototype system, where possible using existing Gallagher technologies, to prove the communication concept illustrated in Figure 1.4.1. The prototype would be integrated in three stages: data acquisition (Figure 1.5.1, [2]), data transmission (Figure 1.5.1, [7]) and radio frequency link (Figure 1.5.1, [3]). The energizer’s logic board (data acquisition) and the SMART PAC (data transmission), which should be used in the prototype, have similar communication capabilities. The prototype could be implemented by using only one of these products to acquire and receive data, however the final prototype both products should be used in order to assess their adaptability to the application. The network functionality (Figure 1.5.1) should be:

  When a message is generated along the electric fence by an energizer (Figure 1.5.1, [1]), the prototype detects it with a logic board (Figure 1.5.1, [2]) and sends it to a transmission RF board (Figure 1.5.1,[3]), which transmits it via RF to another RF board on the farmer’s house (Figure 1.5.1, [4]). The message is transmitted via serial port to the farmer’s PC (Figure 1.5.1, [5]). The message path in Figure 1.5.1 would be: [1], [2], [3], [4] and [5].

In the opposite direction, the farmer's PC generates a message (Figure 1.5.1, [5]), sent through a serial link (Figure 1.5.1, [6]) to the SMART PAC board (Figure 1.5.1, [7]), which will send the message to the fence (Figure 1.5.1, [8]). The message path in Figure 1.5.1 would be: [5], [6], [7] and [8].

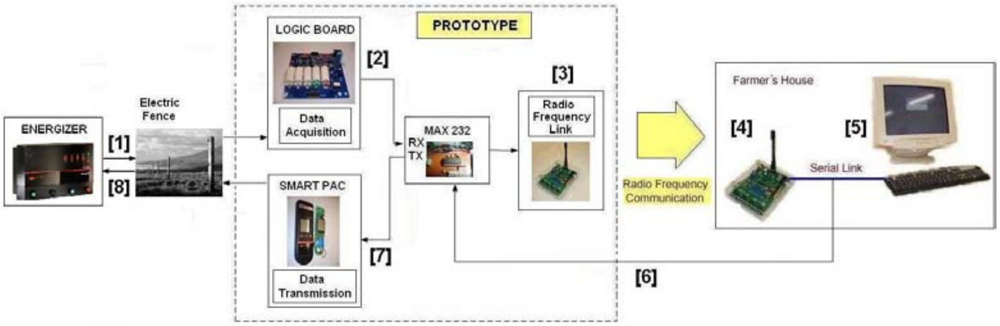


Figure 1.5.1: Final Expected Network

In a future prototype focused on developing a product, the transmission of a message could be done fully wireless as shown in Figure 1.5.2 by modifying the software, however in for this project the link [6] in Figure 1.5.1 should be a serial cable. This makes it possible to evaluate electromagnetic compatibility and noise immunity for systems implemented entirely by serial links and without a radio frequency link.

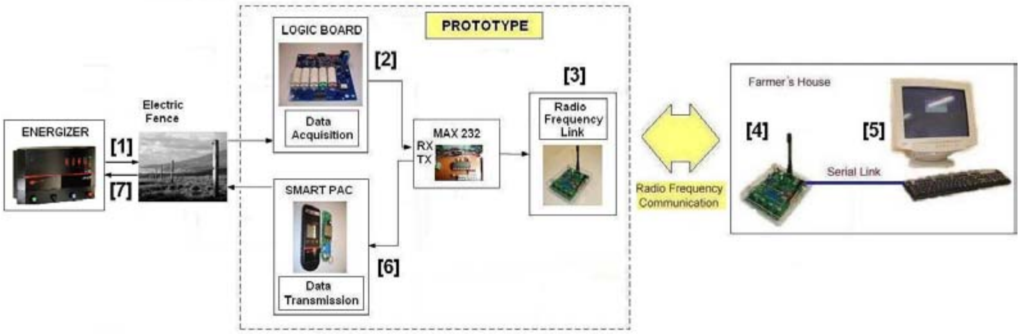


Figure 1.5.2: Network Focused on Developing a Product

- To test the prototype to verify that it fulfils the original technical specifications.
- To evaluate possible solutions to problems and difficulties found during the project's implementation.
- To generate a written report that documents this project to aid in future research.

Chapter 2.
Data Acquisition

2.1 Introduction

This chapter presents the first stage of the prototype that has been constructed for this thesis. It will explain how the so-called “logic board” (commonly used as an energizer’s central processing unit), is used as the device that retrieves information from the fence and transmits it into a radio frequency board. This chapter will also explain the hardware and the software used at this stage. Some details concerning the communication’s format and how the prototype was integrated will also be analyzed in this chapter.

2.2 Data Acquisition Overview

The MX7500 energizer (Figure 2.2.1) is one of the most powerful and efficient energizers in the electric fence market and one of its electronics boards (Figure 2.2.2) (the called “Logic Board”) is used to accomplish an important stage of this thesis.

Figure 2.2.1: MX7500 Energizer

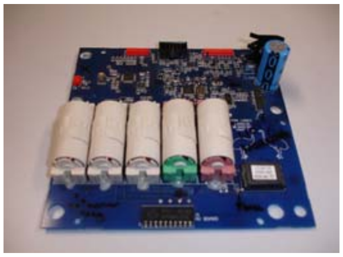


Figure 2.2.2: MX7500 Energizer’s Logic Board

This electronic board was chosen because it has proved to be a practical tool to efficiently acquire data from the electric fence. An important issue in acquiring information from a fence is the retrieval of data from modulated signals traveling along the fence. There are several factors that can make an electric fence a difficult medium through which to transmit data: high voltage pulses generated by an energizer, electromagnetic incompatibility, and impedance changes, among others. To overcome the difficulties associated with these factors and in order to achieve the correct data retrieval, the manufacturer of this board implemented components for protection as well as analog and digital filters. The acquisition and communication components of this board will be used in this project.

2.2.1 Microcontrollers

There are two main electronic components that manage the MX7500 energizer's logic board: a Mitsubishi microcontroller, M30100/M30102 Group and a Texas Instruments digital signal processor (DSP) - microcontroller, TMS320LF2401A (Appendix B.1 Mitsubishi Microcontroller and Texas Instrument DSP).

Although the Mitsubishi IC is the brain of the energizer, it will not be used in the prototype. Its function is to control the entire energizer, including controlling the power circuits that generate the high voltage pulses on the fence, handling the fence's voltage and current readings and displaying functions as well as other tasks that the energizer requires to be carried out.

On the other hand, the Texas instrument DSP, is a 16-bit DSP that takes care of all the energizer's communications with the external world through the electric fence. This DSP-microcontroller is the one that will be used for the prototype application. It is powered at 3.3 V and it is able to perform mathematical functions efficiently. It can operate at 5 MHz or 40 MHz depending on how it is programmed. The DSP is attached to a 10 MHz crystal and has a 10-bit ADC running at 100 kHz.

The communication undertaken between the Mitsubishi IC and the DSP IC makes use of a protocol called XPORT and takes place through the serial port of each device. There is an intermediate hardware stage where the signals are adapted to a 5 V level for the Mitsubishi IC or to a 3.3 V level for the DSP IC.

2.2.2 Communications Overview

In Section 1.3 remote controls for electric fences were introduced. One remote control specifically used in this project is the model SMART PAC that uses a very similar DSP to the one used in the energizer's logic board. Figure 2.2.3 shows, in a very general way, how communications are undertaken between the energizer's logic board DSP and the SMART PAC remote control DSP.

Fence line communication messages are sent along the fence by the SMART PAC remote control. They are received by the energizer at a remote location and instruct the energizer as to how it is to operate. The energizer can also send messages back to the SMART PAC remote control, via the fence.

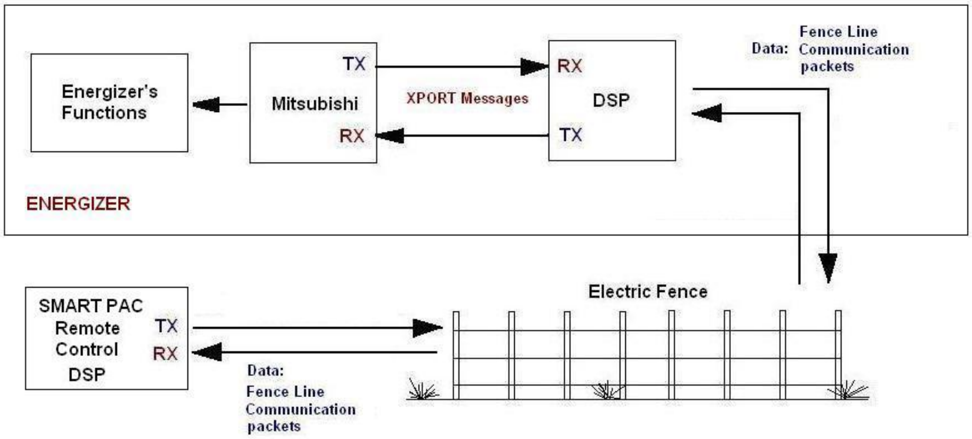

Energizer's Functions
Mitsubishi
TX
RX
XPORT Messages
RX
DSP
TX
Data: Fence Line Communication packets
ENERGIZER
SMART PAC Remote Control DSP
TX
RX
Electric Fence
Data: Fence Line Communication packets

Figure 2.2.3: Communication Between DSPs.

The fence line communications messages transmitted through the fence have are data packets of ten different types (Appendix D: Message Types). These data packets include the necessary information to establish communication between two or more devices.

2.3 Hardware Evaluation

The diagrams described in this section have been retrieved from the MX7500 logic board schematics diagram[7], however most of them have been modified in order to show only the pertinent sections implicated in this project.

2.3.1 Input Circuit

The input circuit that accepts data from a fence is shown in Figure 2.3.1.

There is an array of neon bulbs that are used for protection against high voltage.

[7] Gallagher Group LTD (2005). *MX7500 230V AC Mains Fence Energizers Service Manual*, (ver. 1). Hamilton: Gallagher Group.

The 50 pF input capacitors are placed to reduce the energy of the pulse. This can be achieved since the main frequency components of an energizer's pulse are centered between 1 kHz and 2 kHz and the capacitors at these frequencies have a reactance of 2 MΩ to 3 MΩ, which can dramatically reduce the energy of the pulse. At the communications frequency of around 40 kHz, the capacitors have a reactance of about 85 kΩ (Long M., 2002). These capacitors are also configured to make a resonant circuit with an inductor placed on the communication circuit (Figure 2.3.2) centred at 40 kHz.

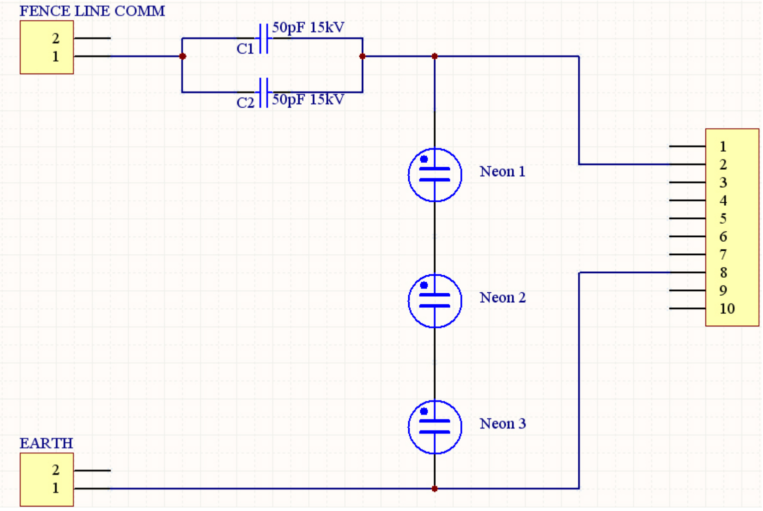


Figure 2.3.1: Input Circuit

2.3.2 Communication Circuit

In the communications circuit, coils are used to provide fence isolation. They reduce the 200 V received signal to a 2 V signal. The modulation method to transport the messages through the fence is known as frequency shift keying (FSK).

For a binary sequence, FSK simply consists of transmitting a single frequency sinusoidal pulse for logic one and a different frequency sinusoidal pulse for a logic zero (Gibson, 1989, p. 150)[8] (Figure 2.3.2).

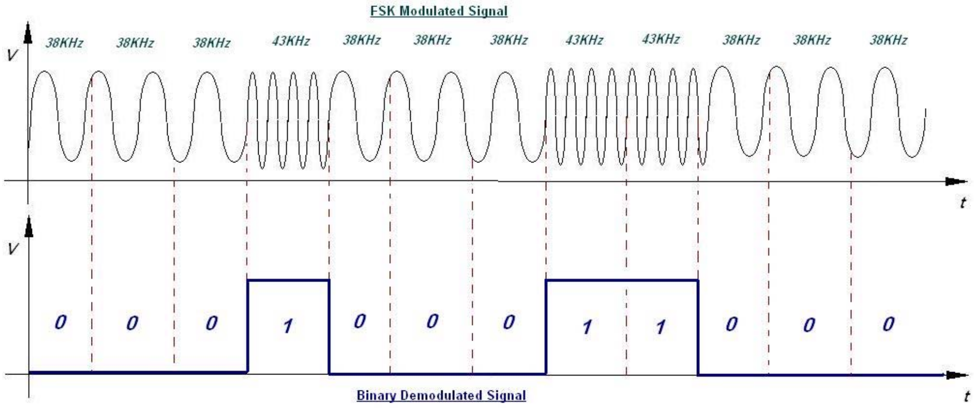


Figure 2.3.2: Frequency Shift Keying Example

A resonant circuit at the input enables the retrieval of the modulated FSK data from the fence. An input inductor L1 (Figure 2.3.3) and the input capacitors C1 and C2 (Figure 2.3.1) form a series resonant circuit tuned at 40 kHz.

> It had been decided to use an FSK system (being one of the most rugged and noise immune systems) centred around 40 kHz. This frequency was chosen for two main reasons. Firstly research had shown that this was a relatively electrically-quiet part of the spectrum. Secondly, frequencies in this range produced a good compromise between the fence and ground characteristics. Basic testing had shown that this sort of frequency could work well for a fence based communications system. (Long, 2002)[9]

[8] Gibson, J. (1989). *Principles of Digital and Analogue Communications*. New York: Macmillan Publishing Company.

[9] Long, M. (2002). *Murray Longs Fence Coupling, Technical Summary.* Hamilton: Gallagher Group.

At the same time, the two input resistors (R1 and R2 in Figure 2.3.3) set the bandwidth from 38 kHz to 43 kHz in order to allow FSK-modulated signals to pass.

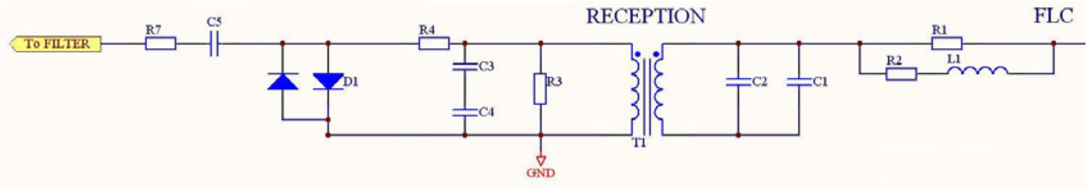


Figure 2.3.3: Communication Circuit

2.3.3 The Pass-Band Filter (Anti-Aliasing Filter)

This filter is made with three operational amplifiers of a specific topology (Figure 2.3.4). The narrow pass-band of the filter rejects all undesirable frequencies (such as those that could be created by the aliasing effect), before they reach the DSP.

The voltage divider at the bottom of Figure 2.3.4 provides a 1.65 V offset to centre the signal on the ADC input range.

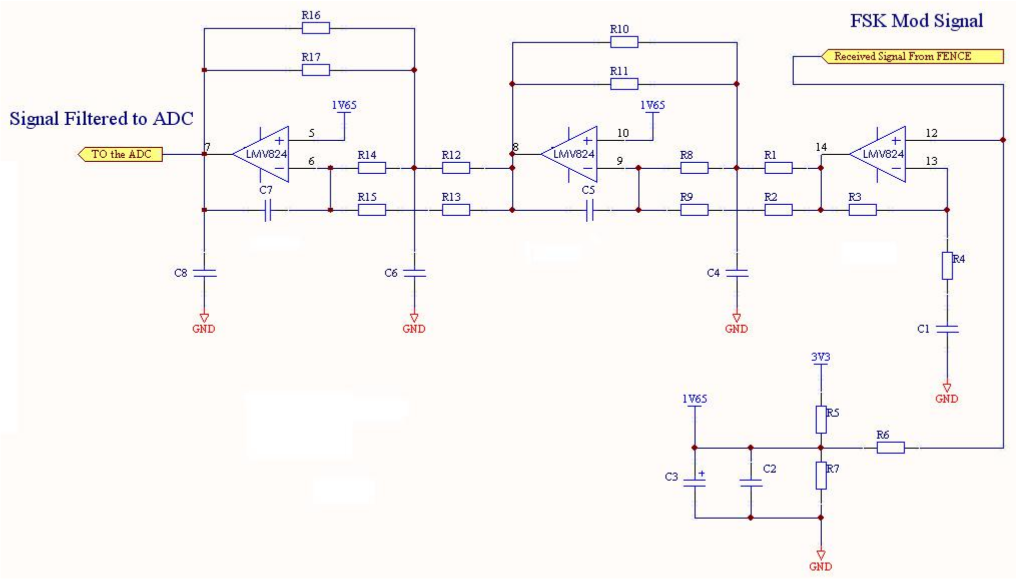


Figure 2.3.4: Pass-Band Filter

2.3.4 The Digital Signal Processor (DSP)

The DSP is a TMS320Lx2401A Texas Instruments and, as previously mentioned, is a hybrid reprogrammable signal processor-microcontroller. It offers a processing performance of 40 million instructions per second (40 MHz). It features a password-based “code security” stored in the internal Flash ROM to avoid unauthorized code duplication. Some of the most relevant features of this DSP are its 10-bit ADC converter and its serial communication interface (SCI) to provide asynchronous communication to other devices.[10]

A Texas Instruments UCC3946 that has a watchdog function holds the DSP in a reset status until the supply voltage rises and remains above the reset threshold for the reset period (Figure 2.3.5)

Therefore the DSP is the core of every data acquisition from the fence and of the transmission of the acquired messages. The DSP has the ability to perform complicated arithmetic functions (for digital filtering), modulations, demodulations and some other communications tasks. In order to accomplish the goals of this project, the DSP communication tasks have been reprogrammed. In the software section some of the previously described functions and programs will be analysed.

[10] Texas Instruments Incorporated (2005). *TMS320LF2401A DSP Controller Datasheet.* Retrieved March 15, 2006 from: http://www.ti.com

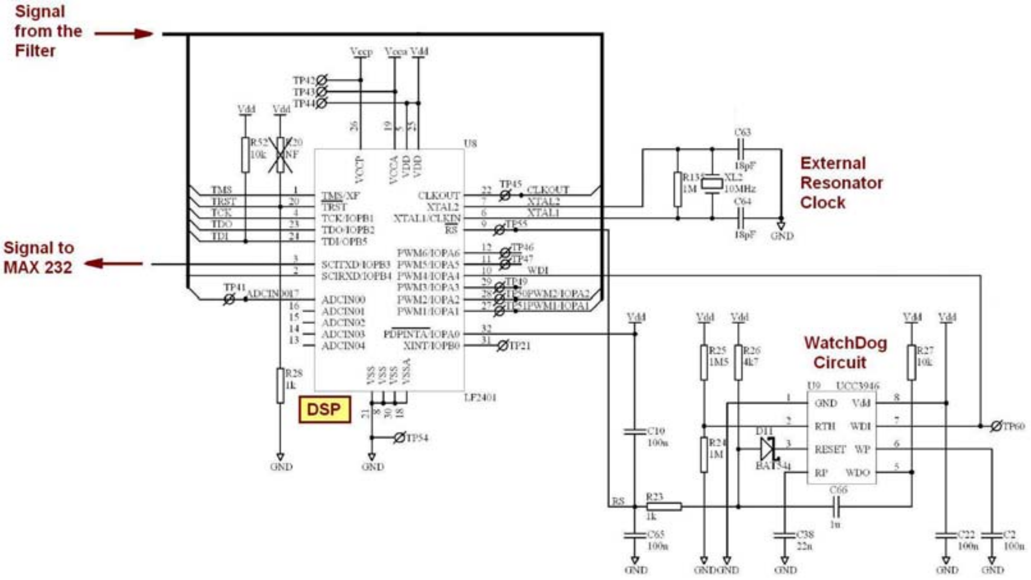


Figure 2.3.5: DSP and Watch Dog Circuit

All the previously described hardware has the purpose of making possible the reception of any FLC packet transmitted along the electric fence. It also sends this received information through the DSP's serial port to the next stage of the project, which is the wireless link.

2.4 Software Evaluation

Even though the hardware requirements have been fulfilled so far, the software must be modified in order to see the desired packet at the serial output port. The original code is particularly complex because the same code is used to program different devices such as the energizer’s logic boards or SMART PAC remote controls. On the other hand, as has been mentioned before, there are modules used to modulate or demodulate and to apply digital filters that should remain in the modified code.

It can be seen in Figure 2.4.1 that at the input there is a hardware filter (Section 2.3.3). After the sampling stage there is a high-pass digital filter tuned at 20 kHz to ensure that the DSP will not handle any harmonic components below 20 kHz. Then there are two band-pass digital filters used for the FSK modulation, which are tuned at 38.10 kHz (the logic 0) and at 42.86 kHz (the logic 1). Most of this digital treatment of the signal is undertaken in a specific module and function called Commstask().

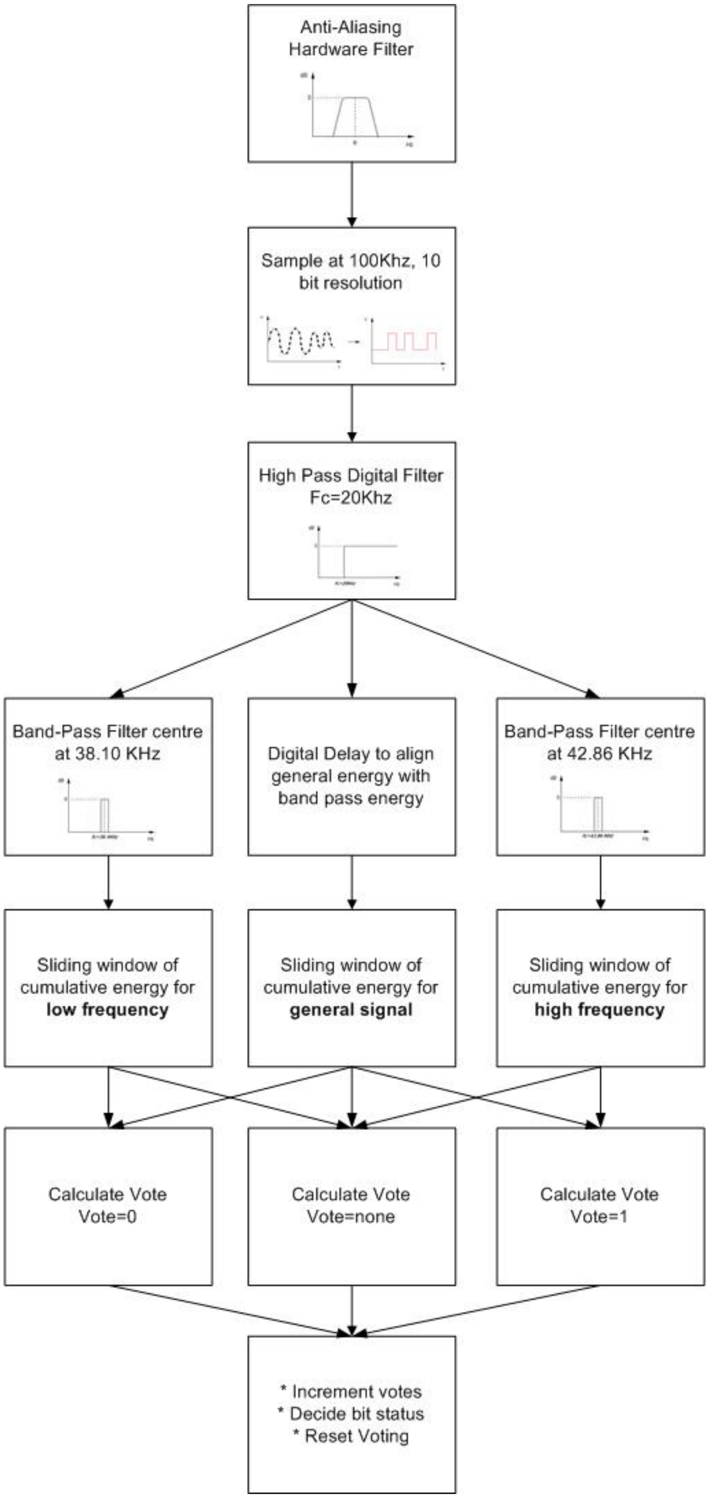

Anti-Aliasing Hardware Filter
Sample at 100Khz, 10 bit resolution
High Pass Digital Filter Fc=20Khz
Band-Pass Filter centre at 38.10 KHz
Digital Delay to align general energy with band pass energy
Band-Pass Filter centre at 42.86 KHz
Sliding window of cumulative energy for **low frequency**
Sliding window of cumulative energy for **general signal**
Sliding window of cumulative energy for **high frequency**
Calculate Vote Vote=0
Calculate Vote Vote=none
Calculate Vote Vote=1
* Increment votes
* Decide bit status
* Reset Voting

Figure 2.4.1: Signal Flow [11]

[11] Body, N. (2002). *SMART PAC FSK Modem Documentation, Signal Flow Graph*. Hamilton: Gallagher Group.

Therefore it is necessary to consider the following chart in Figure 2.4.2 that shows the DSP program flow. Figure 2.4.2 is divided into two sections: the initialization section, where all the initial parameters are set such as watchdogs, memories, state machine stages, communications and ports, and the main loop and the second part integrated by functions that define the main operation loop. The four modules that undertake these tasks are: FSMTask(), TimerTask(), CommsTask() and WatchdogTask() (see Appendix C.1 Acquisition Stage Code). The modulation – demodulation task, the serial communications, as well as the digital filtering are included in the CommsTask() module.

CommsTask() is a public function of the original program and is called once each main loop. It is important to note that the reprogrammed version of the program will be unchanged with the exception of this module. The communication's state machine defined in this function has been reduced into a very simple function, which has the goal of detecting the reception of a valid FLC data packet in order to send it to the serial port output. Figure 2.4.2 [12] shows the original program flow where the CommsTask() function can be seen. A deeper analysis in the communications state machine leads to modifying specific parameters of this function in order to achieve the desired functionality and to respect the integrity of the rest of the code. At this stage it is aimed to invalidate all the communications tasks carried out by the CommsTask() module except for

[12] Body, N. (2005). Functional Flow Diagram SMART PAC – Remote and Fault Finder *High-Level Design* (3rd rev.). Hamilton: Gallagher Group.

the stages involved with packet acknowledgment and packet transmission to the DSP's serial port.

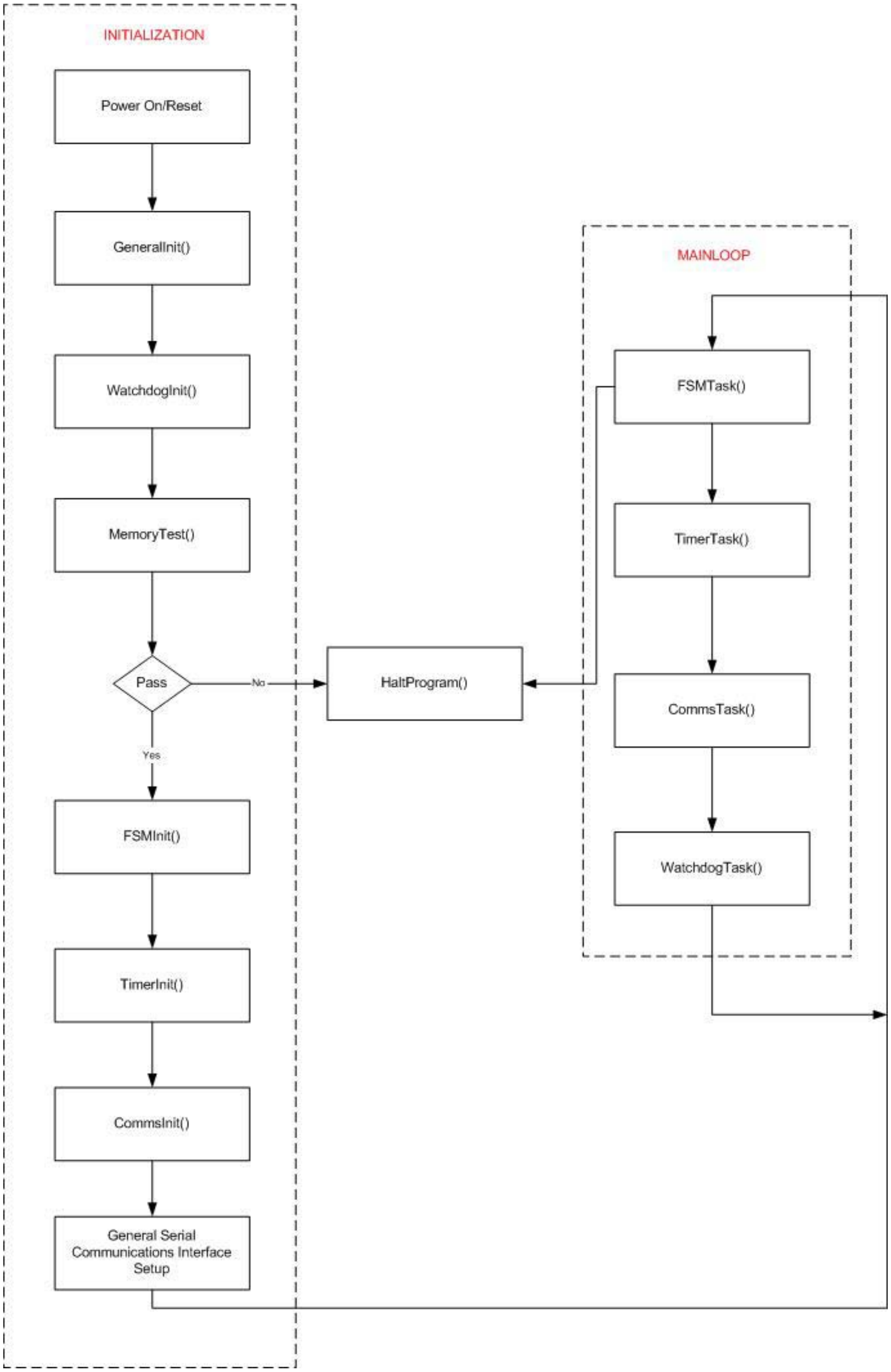


Figure 2.4.2: Logic board's Program Flow Chart

In order to explain how the software modifications were carried out it is appropriate to explain some of the internal components of the CommsTask() function. The function CommsTask() calls GetSMState() a public function which lives inside the FSMTask() module. GetSMState() has the purpose of acknowledging the current state of the communication's state machine. As previously stated, most of these states were erased because their functionality was not relevant for this project. Figure 2.4.3 illustrates the GetSMState() function in a state machine.

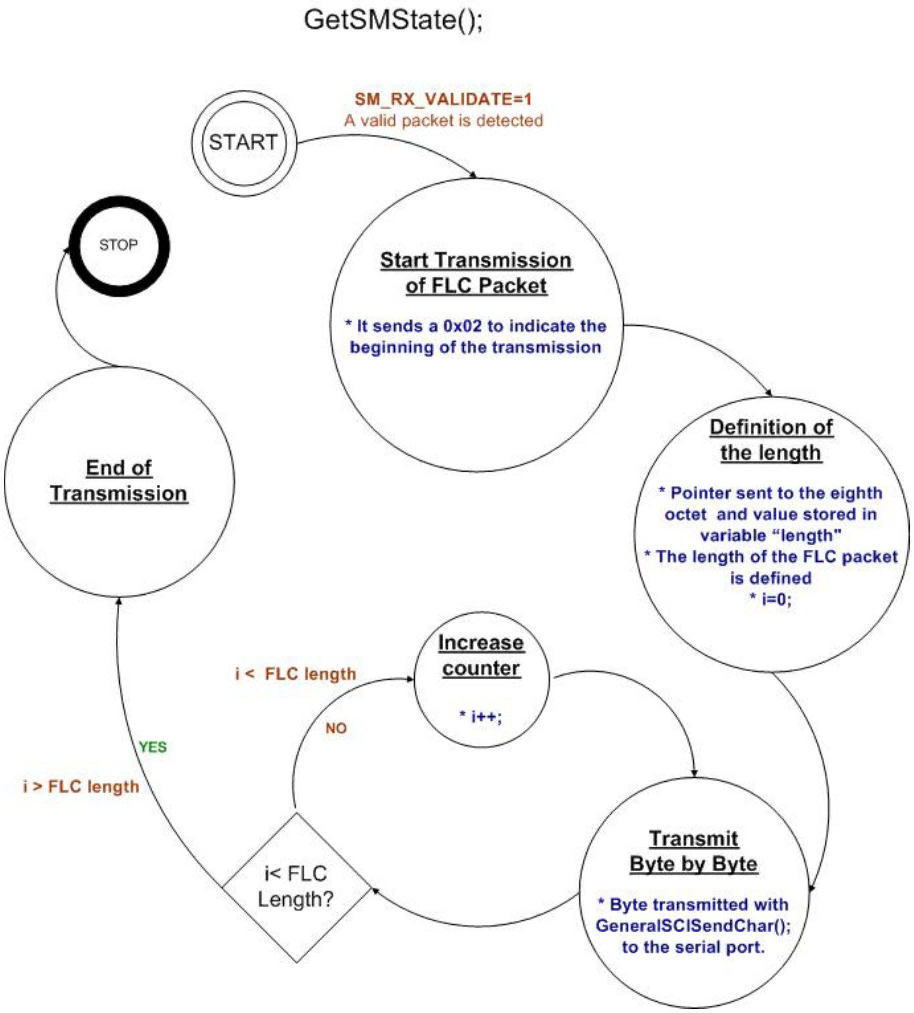

GetSMState();
SM_RX_VALIDATE=1
A valid packet is detected
START
STOP
Start Transmission of FLC Packet
* It sends a 0x02 to indicate the beginning of the transmission
Definition of the length
* Pointer sent to the eighth octet and value stored in variable "length"
* The length of the FLC packet is defined
* i=0;
End of Transmission
Increase counter
* i++;
i < FLC length
NO
YES
i > FLC length
i< FLC Length?
Transmit Byte by Byte
* Byte transmitted with GeneralSCISendChar(); to the serial port.

Figure 2.4.3: Modified Code on GetSMState() Function

Each character is transmitted to the serial port through the function GeneralSCISendChar() in the following way: The process begins when GetSMState () receives a true value from SM_RX_VALIDATE, which indicates the presence of a new message to be transmitted. GeneralSCISendChar() sends each character one-by-one to the serial port. First a 0x02 character is sent to indicate the beginning of transmission. After that, a pointer is sent to the eighth octet (Figure 2.4.4) that contains the length and stores it in the variable “length” in order to record the length of the payload. Then the pointer goes back to the beginning of the FLC packet and a “for” cycle is executed from zero to the length of the payload plus ten. That means the “for” cycle lasts until the length of the payload plus 10 more octets (that is the complete FLC message). The “for” cycle sweeps all the data packet, allowing the GeneralSCISendChar() function to send every character to the serial port.

Once the function GetSMState() was implemented, the main program (Appendix C.1 Acquisition Stage Code) became very simple, however, most of its original functionality (such as initialisations routines) remained unchanged. (Figure 2.4.4)

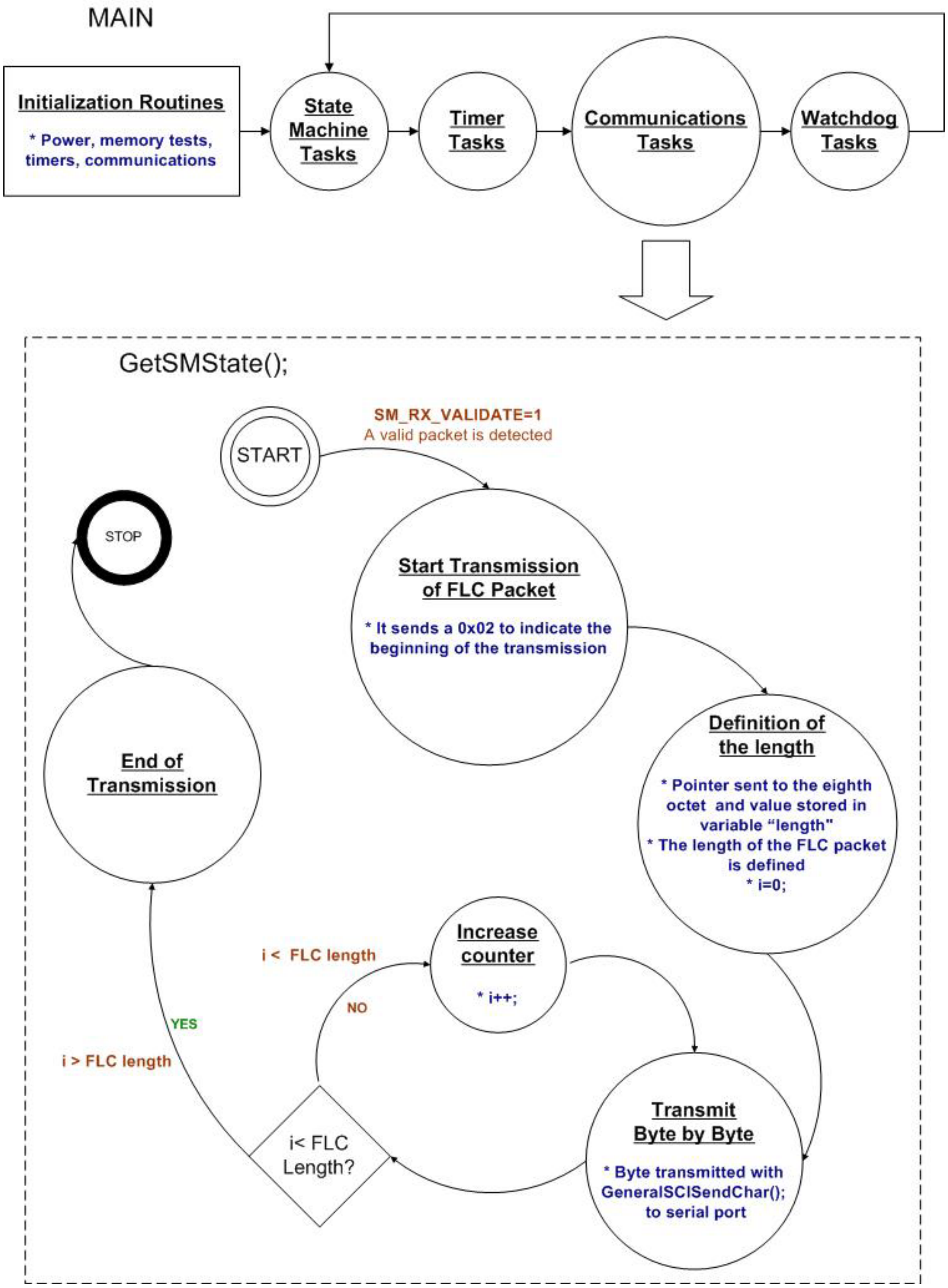

MAIN
Initialization Routines
* Power, memory tests, timers, communications
State Machine Tasks
Timer Tasks
Communications Tasks
Watchdog Tasks
GetSMState();
START
SM_RX_VALIDATE=1
A valid packet is detected
Start Transmission of FLC Packet
* It sends a 0x02 to indicate the beginning of the transmission
Definition of the length
* Pointer sent to the eighth octet and value stored in variable "length"
* The length of the FLC packet is defined
* i=0;
Transmit Byte by Byte
* Byte transmitted with GeneralSCISendChar(); to serial port
i< FLC Length?
NO
i < FLC length
Increase counter
* i++;
YES
i > FLC length
End of Transmission
STOP

Figure 2.4.4: Modified Main Program for the Data Acquisition Stage

2.5 Integration of the Prototype

At this stage, the FLC packet is ready to be taken from the transmission (TX) serial port of the DSP.

In order to implement this stage, the TTL signal level from the DSP's TX serial port was connected to a MAX 232 integrated circuit in order to send it via RS-232 levels to the next stage, which is the wireless link.

2.5.1 Tests

The tests consisted of sending messages from a SMART PAC to the prototype's logic board. The SMART PAC output was connected to the logic board's input. Then "On", "Off" or "Inquire" messages were transmitted. Subsequently the packets were received and displayed on a PC with a "HyperTerminal" software (which allowed to see the incoming characters from the serial port) set at 9,600 bps.

This is an example of the characters transmitted in an OFF Message:

```
6A D8 7B 6A D3 77 47 07 02 C6 E4 DE 9D 03 D6 EC 06
```

The payload in the messages is encrypted.

2.6 Summary of Results

This chapter describes how the FLC data packets are retrieved from the electric fence in order to become the "payload" of the following communication stages. It also explains the most relevant details of the hardware and software involved in this process, as well as their integration into the prototype.

Chapter 3. Data Transmission

3.1 Introduction

The data acquisition stage is exposed in this chapter by introducing concepts concerning the transmission line theory and the monitoring device used for implementing the prototype. The hardware and software of this device are also analysed in order to define the required changes.

3.2 Line Transmission Theory

Line transmission theory describes the basis of transmission of electrical energy from one place to another. In some cases the energy can be very low, for example in applications such as communications and signal measurement. This thesis is concerned with communications using line transmission theory where the energy to transmit is very low. (Figure 3.2.1).

Figure 3.2.1: Parallel Conductor Lines Configuration

3.2.1 The Principle

Wires of an electric fence can propagate electric waves along its length. What are known as “transverse electromagnetic waves” or “principal waves” require two or more conductors for their existence. These waves propagate along the length of conductors and their electric and magnetic fields travel transversely to their direction of propagation. This kind of wave is very similar to a plane wave and can be used to transmit signals along transmission lines (Guru & Hiziroglu, 1998, pp. 367-432)[13]. The configuration used for the electric fences is known as “parallel-round conductor lines” (Figure 3.2.2, reproduced from: Guru & Hiziroglu,1998).

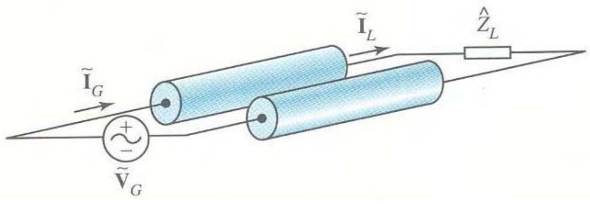


Figure 3.2.2: Parallel-Round Conductor Transmission Line

3.2.2 Reflections Through Transmission Lines

Electric fences are far from being an ideal communication medium. Discontinuities are often present along the length of the fence for reasons difficult to control, such as growing grass, fallen trees or branches. In electrical terms, a discontinuity point is a place along the transmission line where the characteristic impedance of the line changes.

[13] Guru, B.S. & Hiziroglu, H.R. (1998). *Electromagnetic Field Theory Fundamentals.* Boston: PWS Publishing Company.

Figure 3.2.3 (reproduced from Guru & Hiziroglu,1998), shows a transmission line with a change in impedance modelled as a load impedance Zl.

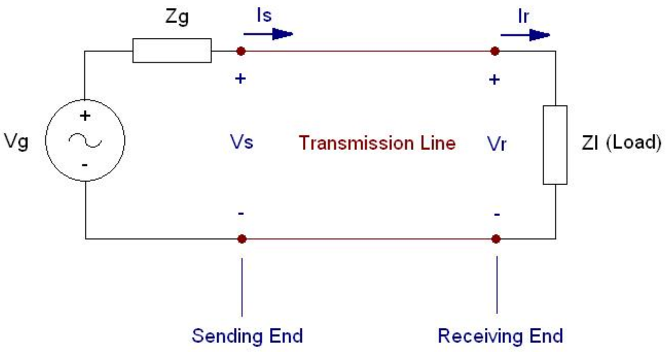


Figure 3.2.3: Model of a Transmission Line.

A wave is reflected when it finds a considerable change of impedance (discontinuity point) that forces it to find its way back and it is transmitted beyond the discontinuity point until it reflects back to the source (Figure 3.2.4) (Staelin, Mortgenthaler, & Kong, 1994, p. 220)[14].

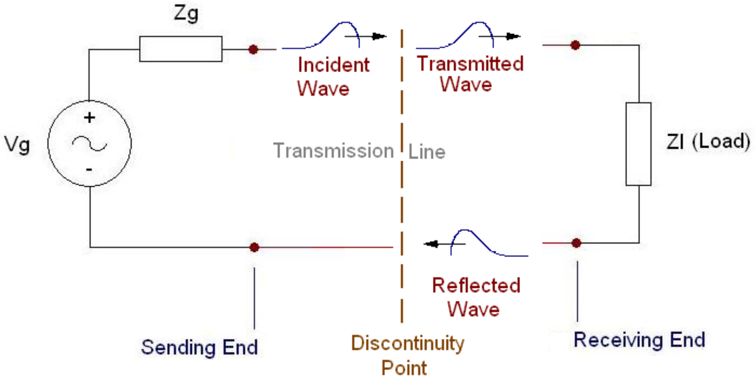


Figure 3.2.4: Reflected Wave

[14] Staelin, D.H. , Mortgenthaler, A.W. & Kong, J. A. (1994). *Electromagnetic Waves.* New Jersey: Prentice Hall.

This concept was considered during the early stages of the electric fence's faultfinders such as the SMART PAC remote control, which can evaluate impedance changes to carry out troubleshooting tasks using the electric fence as a communication medium.

3.3 SMART PAC Board

Research has helped to develop products such as faultfinders and troubleshooting devices that have successfully satisfied the needs of the electric fence market. The SMART PAC remote control (Figure 3.3.1) has been used for the transmission stage of this project. Although its hardware and software share the same principles as the logic board analysed in chapter 2, it is important to note that the SMART PAC was mainly designed to transmit and receive information.

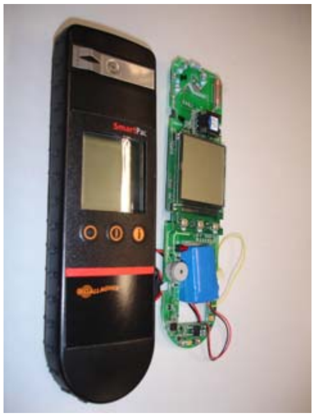


Figure 3.3.1: SMART PAC Remote Control and its Electronic Board

3.4 Hardware Description and Evaluation

This is a similar evaluation to the one undertaken in the previous chapter for the energizer's logic board. However since most of the components are very similar to the ones used by the energizer's logic board, it is intended only to comment on the most relevant of them. These devices can be divided in three sections: power, communications and the DSP. The purpose of this hardware is to make the final prototype able to receive FLC packets or any characters coming from a PC's serial port and send them into an electric fence transmission line.

3.4.1 Power

In normal operation conditions, a 9 V rechargeable battery powers the SMART PAC remote control, however the final prototype was powered with a regulated 9 V power supply.

3.4.2 JTAG Port and Serial Port

The SMART PAC DSP was reprogrammed from a port called JTAG that was originally designed for that purpose. On the other hand, the serial communication established with the PC is carried out through the DSP's serial port shown in Figure 3.4.1. The serial link coming from the PC was first connected to a MAX 232 circuit in order to adapt the voltage to the DSP's serial port. Ideally the information transmitted from the PC to the fence should be a FLC packet, so that the message can be acknowledged by another device, however any packet format can be sent to the fence.

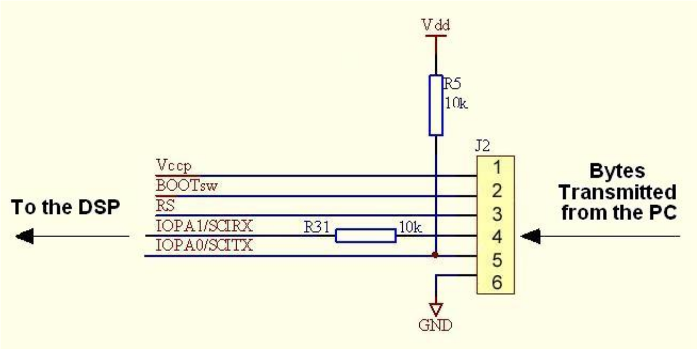


Figure 3.4.1: SMART PAC & DSP's Serial Port

3.4.3 Transmission Circuit

For this application two main circuits are used from the SMART PAC remote control to incorporate the transmission section: the fence coupling circuit and the H-bridge driver. The fence coupling circuit creates the resonance needed to generate the FSK signals and the H-bridge drives the DSP signals out to the fence via the transformer (Figure 3.4.3).

3.4.3.1 Transmission Circuit: H-bridge Driver

The H-bridge driver (Figure 3.4.2) consists of four field effect transistors, which are driven with 3.3 V or 0 V that reproduce the signals provided by the DSP. The four FETs form two circuits; one side is connected to the battery voltage and the other to the ground. When the transmitter is not operational, the FET p-channels are put to logic ‘1’ and the n-channels to logic ‘0’, which identifies them as an “OFF” state. From the fence side, the drains will look like being at high impedance.

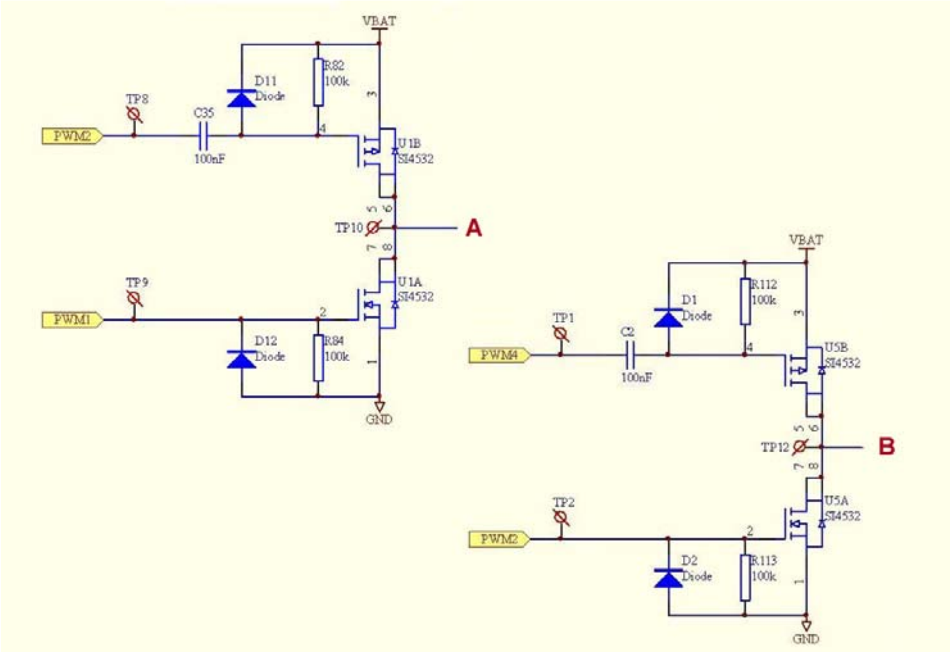


Figure 3.4.2: H-Bridge Driver

3.4.3.2 Fence coupling circuit

The fence coupling circuit (Figure 3.4.3), consists of a resonant circuit across the points A and B (coming from the H-bridge driver, Figure 3.4.2) that drives the FSK modulated signal onto the fence through the transformer. The neon bulbs protect the circuit from high voltage frequency spikes generated in the fence (Long, 2002).

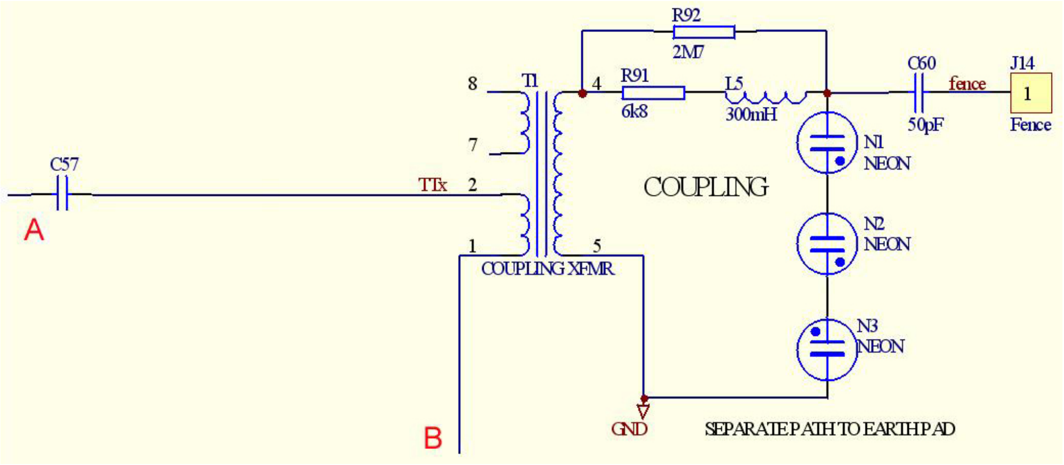


Figure 3.4.3: Fence Coupling Circuit

3.4.4 The SMART PAC's DSP

The DSP used on a SMART PAC remote control is the TMS320LF2403A (Appendix B.1 Mitsubishi Microcontroller and Texas Instrument DSP). It is a 16-bit hybrid DSP-microcontroller; powered at 3.3 V and it can be programmed to run at 5 MHz or 40 MHz. The crystal and the phase lock loop (PLL) circuits (Figure 3.4.4), control the DSP's clock. By internally multiplying the crystal speed of 10 MHz, times 0.5 or 4 with the PLL circuit, the DSP's can run at 5 MHz or 40 MHz. Its 10-bit ADC runs at 100 kHz. This DSP is very similar to the one used in the logic board of chapter 2 (TMS320LF2401A), however, since the SMART PAC is a portable more independent device and it has more interaction with external devices such as a liquid crystal display (LCD) and a keypad, it requires additional inputs and outputs, which are provided by the TMS320LF2403A DSP.

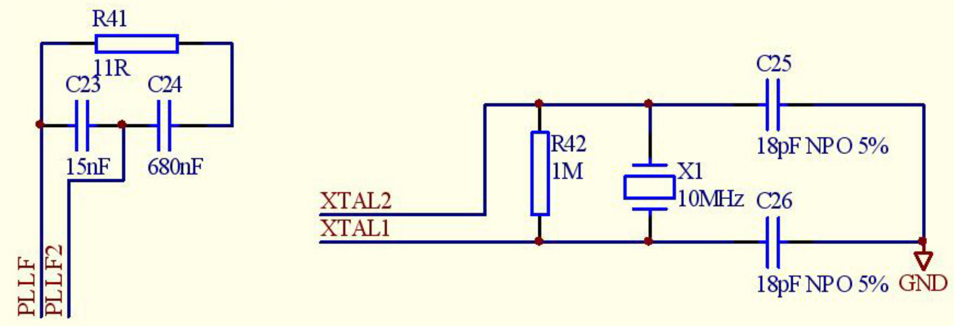


Figure 3.4.4: Crystal and PLL

3.5 Software Evaluation

At this stage the only changes to the SMART PAC were carried out at a software level. Although the original code is very similar for both devices (energizer's logic board and SMART PAC remote control), the final programs were different and had different functions. Figure 3.5.1 shows the main program flow, where additional functions such as KeyTask(), BatteryTask(), LcdTask() and BeeperTask() are used. However none of these functions were modified and only the communication module was reprogrammed.

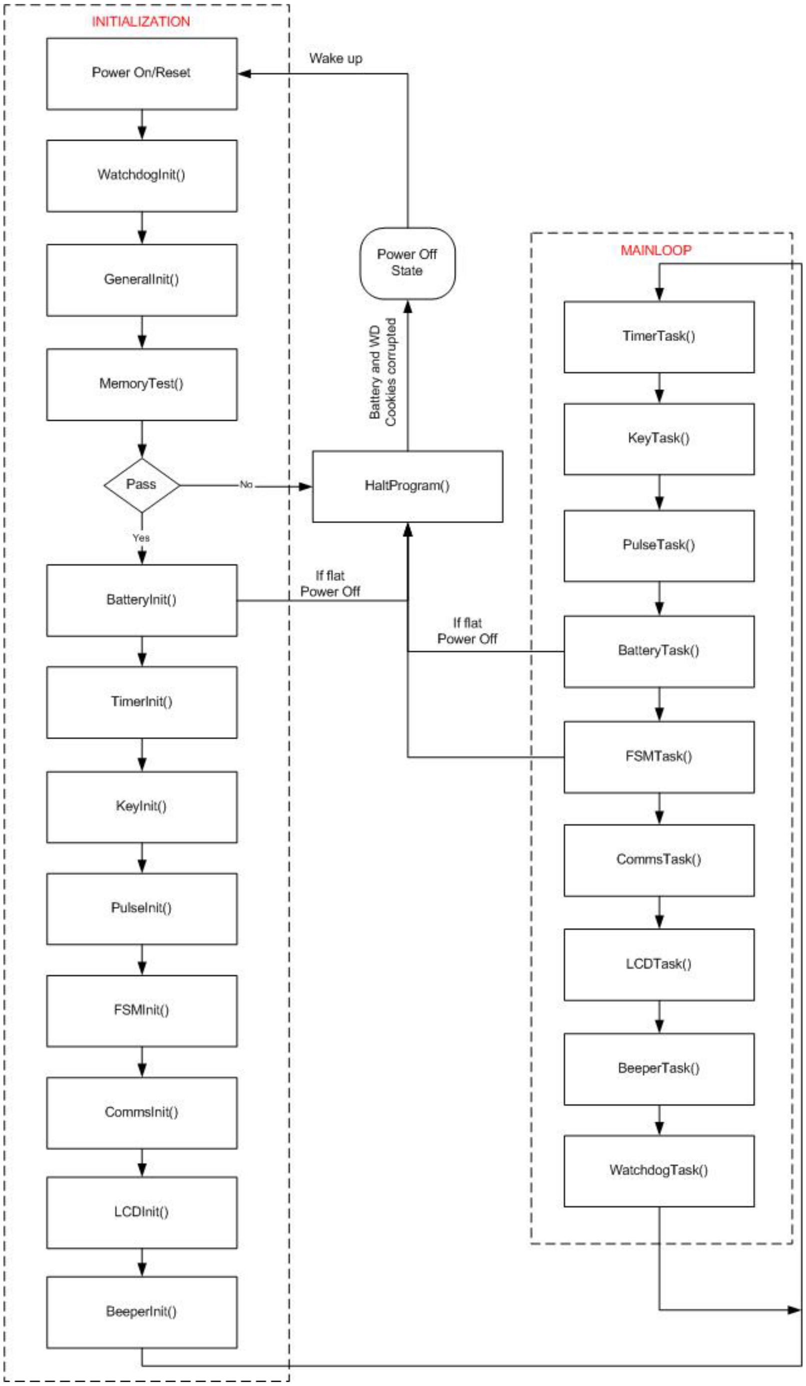


Figure 3.5.1: Software Functional Flow for the SMART PAC

Following the same approach of chapter 2, the CommsTask() function was modified in this chapter. The following Figure 3.5.2 (Body, 2002) shows the SMART PAC communications state machine.

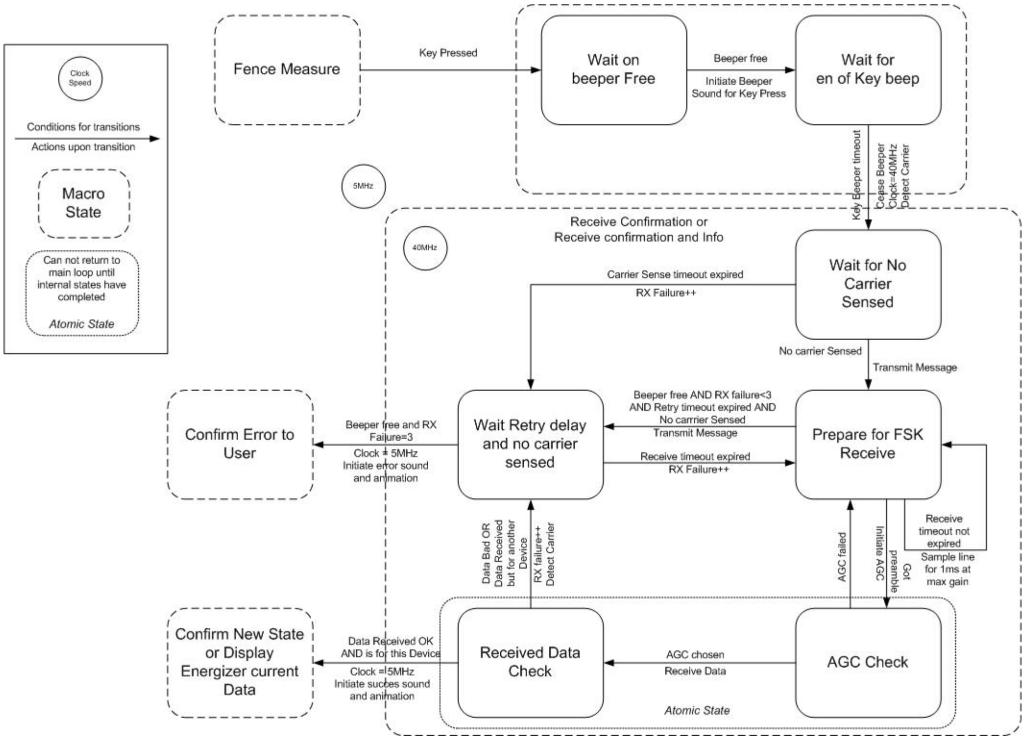
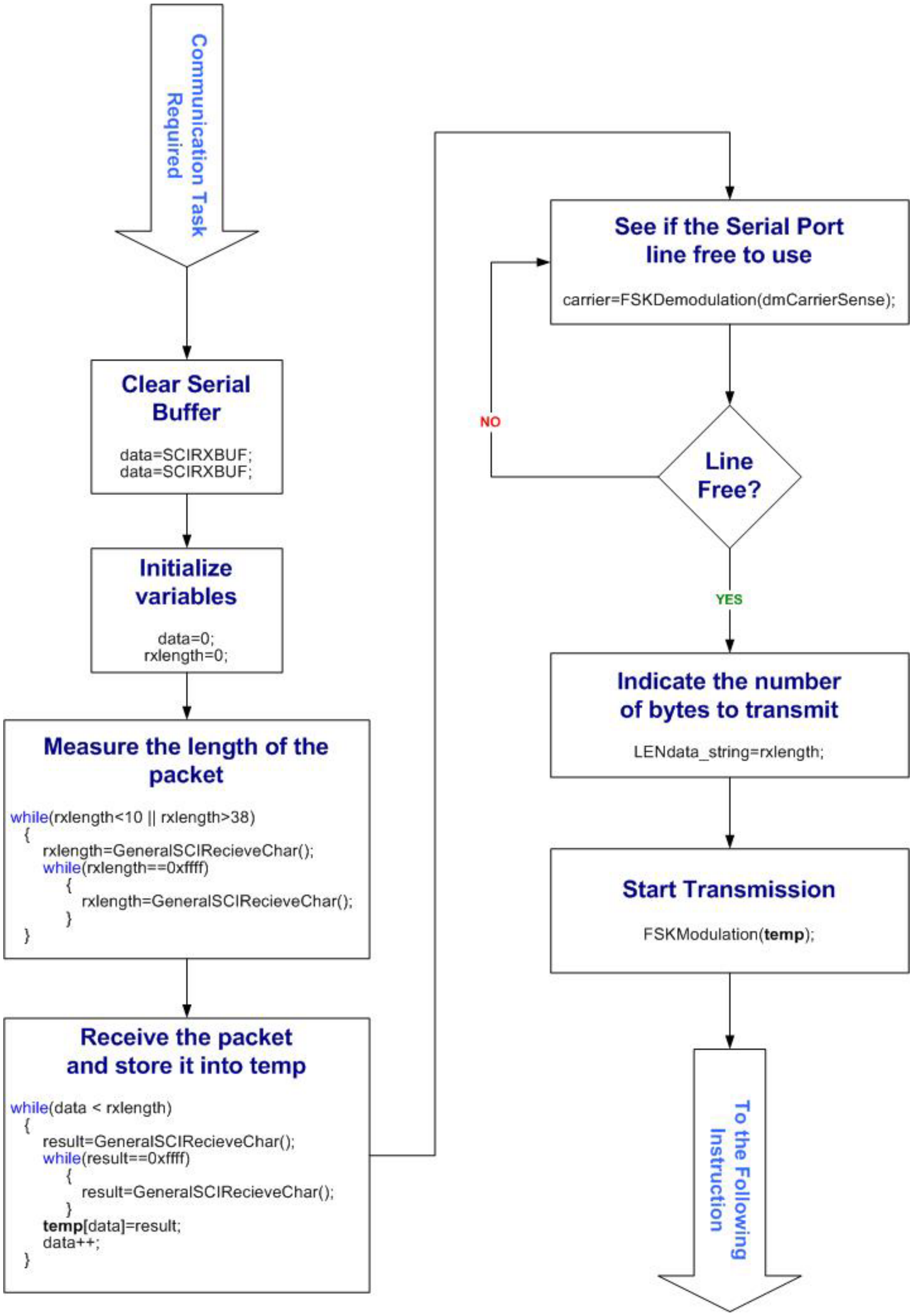

Clock Speed
Conditions for transitions
Actions upon transition
Macro State
Can not return to main loop until internal states have completed
Atomic State
Fence Measure
Key Pressed
Wait on beeper Free
Beeper free
Initiate Beeper Sound for Key Press
Wait for en of Key beep
Key Beeper timeout
Cease Beeper
Clock=40MHz
Detect Carrier
5MHz
40MHz
Receive Confirmation or Receive confirmation and Info
Wait for No Carrier Sensed
Carrier Sense timeout expired
RX Failure++
No carrier Sensed
Transmit Message
Confirm Error to User
Beeper free and RX Failure=3
Clock = 5MHz
Initiate error sound and animation
Wait Retry delay and no carrier sensed
Beeper free AND RX failure<3 AND Retry timeout expired AND No carrrier Sensed
Transmit Message
Receive timeout expired
RX Failure++
Prepare for FSK Receive
Receive timeout not expired
Sample line for 1ms at max gain
Data Bad OR Data Received but for another Device
RX failure++
Detect Carrier
AGC failed
Initiate AGC
Got preamble
Confirm New State or Display Energizer current Data
Data Received OK AND is for this Device
Clock = 5MHz
Initiate succes sound and animation
Received Data Check
AGC chosen
Receive Data
AGC Check
Atomic State

Figure 3.5.2: SMART PAC State Machine

The function called CommsTask(), defines most of these communications states. However, since the requirements of this stage are simply to retrieve the packets of characters arriving from the serial port and send them through to the fence, most of the original states defined in this state machine were removed. Unused code was removed, and these modifications led us to the final required behaviour. The desired behavior reprogrammed in the CommsTask() function is described in Figure 3.5.3.


Communication Task Required
Clear Serial Buffer
Initialize variables
Measure the length of the packet
Receive the packet and store it into temp
See if the Serial Port line free to use
Line Free?
NO
YES
Indicate the number of bytes to transmit
Start Transmission
To the Following Instruction

```
data=SCIRXBUF;
data=SCIRXBUF;
```

```
data=0;
rxlength=0;
```

```
while(rxlength<10 || rxlength>38)
  {
    rxlength=GeneralSCIRecieveChar();
    while(rxlength==0xffff)
        {
          rxlength=GeneralSCIRecieveChar();
        }
  }
```

```
while(data < rxlength)
  {
    result=GeneralSCIRecieveChar();
    while(result==0xffff)
        {
          result=GeneralSCIRecieveChar();
        }
    temp[data]=result;
    data++;
  }
```

```
carrier=FSKDemodulation(dmCarrierSense);
```

```
LENdata_string=rxlength;
```

```
FSKModulation(temp);
```

Figure 3.5.3: Event Flow for Commstask() Function

3.6 Integration of the Prototype

The hardware integration on the prototype was undertaken by powering the required 9 V on the board and connecting the SMART PAC DSP receive (RX) serial port pin to the MAX 232 Circuit. Therefore the messages transmitted from the PC were received by the prototype, which sent them to the fence (Figure 3.6.1).

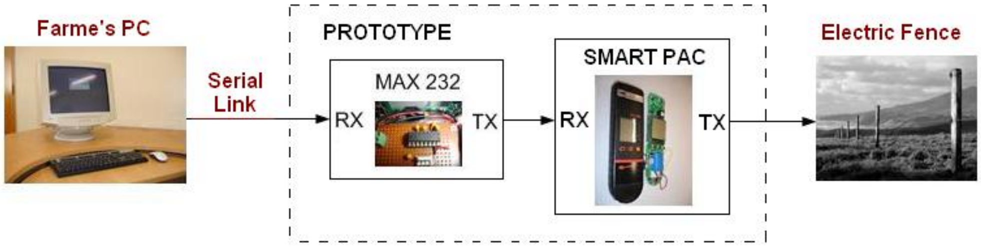


Figure 3.6.1: Prototype Integration

3.6.1 Tests

FLC packets (on, off or inquire messages) were sent from the PC to the prototype with the "HyperTerminal" software, which allows seeing incoming characters and transmitting them through the serial port as well. The SMART PAC board received the packets sent from the PC and sent them to the prototype's logic board input, described in chapter 2. Then the energizer's logic board acknowledged the packets and displayed them again with HyperTerminal through a different serial port.

The SMART PAC remote control beeper indicated that the first packet had been sent. The following packets did not produce any sound on the beeper, as expected.

3.7 Summary of Results

The prototype is now integrated with the energizer’s logic board (for reception) and the SMART PAC remote control board (for transmission).

At this stage it is possible to establish communication in both directions. The following chapter explains the wireless radio frequency solution applied and the final prototype implementation.

Chapter 4.
Radio Frequency Link

4.1 Introduction

The radio frequency link represents the core of this project. Some general guidelines were followed in order to define the technology to use. In this chapter the radio frequency hardware is chosen, as well as its software program and are analysed in order to fulfil the appropriate functionality of the final prototype.

4.2 Wireless Radio Frequency Link

The essential concern about a radio frequency link is the reliability of the transmission and reception process. Some useful parameters[15] to determine radio frequency link reliability were considered during the initial stages of this project, in order to find the most adequate developer kit to use. These are:

- Sensitivity

  Sensitivity is a measure of the receiver's ability to detect signals usually indicated in microvolts (Carr, 2001, p. 219)[16]. In other words it is the minimum received power (-100 dBm for the Nordic Transceiver) for an acceptable bit error rate during reception.

[15] Karlsen, F. (2000). *Guidelines to low cost wireless system design.* Retrieved October 31, 2006 from: http://www.nordisemi.no/

[16] Carr, J.J. (2001). *Secrets of RF Circuit Design* (3rd ed.). New York: Mc Graw Hill.

- Output power and operating frequency

The following Table 4.2.1[17] shows the spectrum allocation of the operation frequencies of the prototype described in this thesis, as well as the power allowed for each frequency.

Table 4.2.1 : Frequencies and Power

From (MHz)	To (MHz)	Power (mW)	Designated Use
433.05	434.79	25	Telemetry
864	868	1000	Unrestricted
915	921	3	Telemetry
921	929	1000	Unrestricted

None of these frequencies need telecommunications authority approval in New Zealand. In Europe, the relevant frequencies go from 433 to 434.79 MHz and are attached to the standard EN 301 783, which does not require a telecommunications authority approval[18]. The International Telecommunications Union (ITU) (Losee, 2005, p. 23)[19] provides frequency allocations on a worldwide basis; however, special considerations should be taken made for each country since radio frequency regulations can vary. For the Nordic Transceiver these frequencies are configurable.

- Adjacent channel selectivity (ACS)

ACS is a measure of the receiver's ability to demodulate a signal,

[17] NZ Gazette (2003). *Notice: go3206, Radio Communications Regulations*, Retrieved April 17, 2006 from: http://www.secnet.co.nz/

[18] ERO (2006). *The European Table of Frequency Allocations and Utilisations. Report 25*. Retrieved April 17, 2006 from: http://www.ero.dk/

[19] Losee, F.A. (2005). *RF Systems, components, and circuits handbook,* (2nd ed.). Massachusetts: Artech House.

with the presence of a sine component centred in the adjacent channel[20]. The Nordic IC provides an ACS attenuation of -16 dB.

- Modulation principle

  The modulation principle for the Nordic transceiver is the gaussian frequency shift keying (GFSK), which is an enhanced version of frequency shift keying. The goal of GFSK modulating principle is to allow a higher bit rate to be transferred (with the same bandwidth). This is achieved by filtering the data through a Gaussian filter before undertaking the carrier modulation, resulting in a reduction of the power spectrum.

- Circumstantial factors

  Floors, walls, buildings and windows must be considered to account for losses in a radio frequency transmission. (Table 4.2.2)

Table 4.2.2 : Losses Caused by Transmission Obstacles

Object Causing Path Loss	Typical Loss [dB]
Wall (indoor)	10-15
Wall (exterior)	2-38 (percentage of windows and height important)
Floor	12-27
Window	2-30 (metal tinted windows cause high loss)

[20] ETSI European Telecommunications Standards Institute (1999). *Radio Transmissions and Reception, UMTS XX.06 Version 1.0.0,* Retrieved April 17, 2006 from: http://www.3gpp.org/ftp/tsg_ran/WG4_Radio/TSGR4_01/Docs/Others/P-99-097.doc

4.2.1 Developer's Kit

According to the previously described criteria, different technologies were considered to implement the application. However, it is difficult with only theoretical assumptions to be sure about the limitations and advantages of a specific technology. For this reason, three brands were considered NORDIC, ATMEL and CHIPCON. Their technology and products are very similar and the greatest differences between them are more related to their developer tools (such as software debugging tools) rather than radio frequency reliability parameters. The chosen brand to develop this application was Nordic. The Nordic nRF9E5 evaluation or developer kit fulfils the basic needs of this project, such as ease of programming, featured devices such as the Nordic nRF9E5 IC with transceiver and embedded 8051 compatible microcontroller, EEPROM memory, a crystal oscillator and a serial output port.

4.2.2 Early Stages Test

The evaluation kit (that included two RF boards) was originally tested to assess its range. An example application included in the kit was programmed to each board, one of them working as a receiver and the other as a transmitter. The code allowed the transmission of a signal from one board to the other by pressing the board's push buttons. The signal was transmitted from the transmitter board and then sent to the receiver board that acknowledged the reception by lighting an LED. The range tests were done in open fields with different weather conditions, considering obstacles such as concrete walls and windows. The maximum transmission range without obstacles on a sunny day was of about 500 m, therefore the maximum range required for the project (Section 1.4.1) was reached.

4.3 Hardware Evaluation

The core of the Nordic NRF9E5 evaluation board is the single microchip nRF9E5 transceiver with an embedded 8051 compatible microcontroller and a 4-input 10-bit ADC (Appendix B.2 Data Sheet for Details of the nRF9E5 Transceiver).

The following Figure 4.3.1 shows on a snapshot of the board and the position of some important circuits used from this evaluation board.

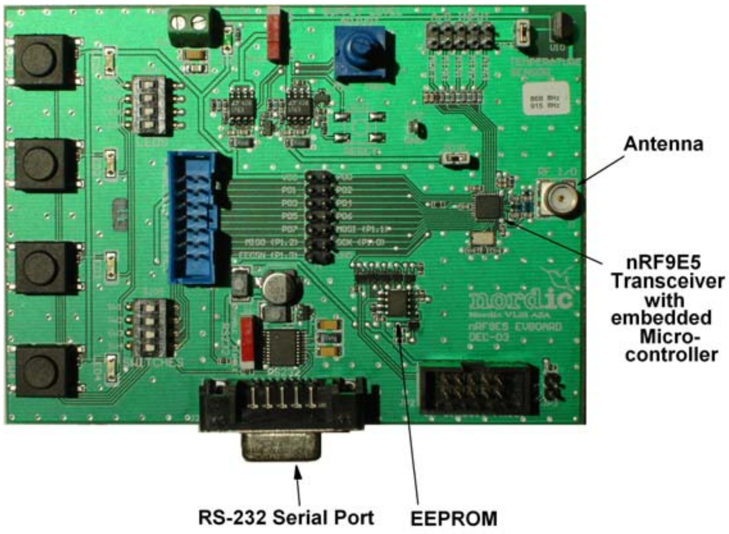


Figure 4.3.1: Nordic nRF9E5 Evaluation Board

The evaluation board incorporates peripherals such as:

- Connectors to ports
- Multi-purposed buttons with LEDs linked to four ports
- A 4.5 to 12 V DC power supply

- A MAX 232 IC with RS232 connector
- 4 input 10-bit analog to digital converter (ADC)
- A USB port for loading the user program

In this subsection, only the relevant circuits and peripherals concerning this project will be analysed in order to describe how the different sections integrate into the final prototype.

4.3.1 The nRF9E5 Radio Transceiver

The radio transceiver consists mostly of a frequency synthesizer, a power amplifier, a modulator, and a receiver unit (Appendix B.2 Data Sheet for Details of the nRF9E5 Transceiver).

The synthesizer is based on a frequency comparison circuit in which the output is compared in frequency and in phase to the output of the crystal's fixed frequency reference. If a difference is detected between the two compared frequencies, or if any phase difference is seen between the two oscillating signals, the phase locked loop (PLL) automatically compensates by moving output frequency or phase up or down until both oscillators are locked at the same frequency and phase (Lenk, 1999, p16)[21]. The transceiver's output power and frequency are programmable and can operate at 433, 868 or 915 MHz. However there are also some hardware modifications to implement if the frequency is to be modified. Table 4.3.1 shows the component values that must be modified from the schematic diagram on Figure 4.3.3 to change the operating frequency.

[21] Lenk, J.D. (1999). *Optimizing Wireless / RF Circuits*. New York: McGraw Hill.

Table 4.3.1 : Component's Values to Modify the Frequency

Component	868 / 915 MHz	433 MHz
C3	33 pF, ± 5%	180 pF, ± 5%
C9	3.9 pF, ± 0.25 pF	18 pF, ± 5%
C10	3.9 pF, ± 0.25 pF	18 pF, ± 5%
C11	Not Fitted	Not Fitted
C12	33 pF, ± 5%	6.8 pF, ± 5%
C13	Not Fitted	Not Fitted
L1	12 nH, 2%	12 nH, 2%
L2	12 nH, 2%	39 nH, 5%
L3	12 nH, 2%	39 nH, 5%

Modifications in the software must also be undertaken to set the appropriate operating frequency. Configuring registers CH_NO and HFREQ_PLL in the program set the operating frequency as is shown in Equation 4.3.1 [22] :

$$f_{OP} = (422.4 + (CH\_NO/10))(1 + HFREQ\_PLL)\ MHz$$

Equation 4.3.1: Output Operating Frequency of nRF9E5 IC

As an example, in Table 4.3.2 the three required frequencies for this project are calculated. In this application the frequency was programmed at 868 MHz.

Table 4.3.2 : Examples of Operating Frequencies

Operating Frequencies	HFREQ_PLL	CH_NO
433.1 MHz	0	001101011 (dec: 107)
868.4 MHz	0	001110110 (dec: 118)
927.8 MHz	1	110011111 (dec: 415)

[22] Nordic Semiconductors (2004). *Product Specification nRF9E5* (1.1 rev.). Retrieved October 31, 2006 from: http://nordicsemi.no/

4.3.2 Modulation

As previously stated, the modulation of the nRF905 is GFSK with a data rate of 100 kbps.

A conventional FSK-modulated signal is generated by shifting the carrier to reflect the digital information that has been transmitted. Switching from one frequency to another can be accomplished by having some separated oscillators tuned to the desired frequencies and selecting one of those frequencies. This quick switching from one oscillator output to another in successive signalling intervals results in lobes in a large spectral side outside of the main spectral band for transmission of the signal (Proakis, 2001, pp. 185-189)[23]. This availability of useful spectrum is important when bandwidth limitations exist; such as occurs in this application where GFSK modulation results in a more useful bandwidth transmission link.

4.3.3 The Embedded Microcontroller

The microcontroller is a DW8051 Macro-Cell manufactured by Synopsys Corporation. Its hardware features and instruction cycle timing are similar other 8051 microcontrollers. The following Figure 4.3.2 (Nordic, 2004) shows how the memory of this microcontroller is organized.

[23] Proakis, J.G. (2001). *Digital Communications* (4th ed.). Boston: McGraw Hill.

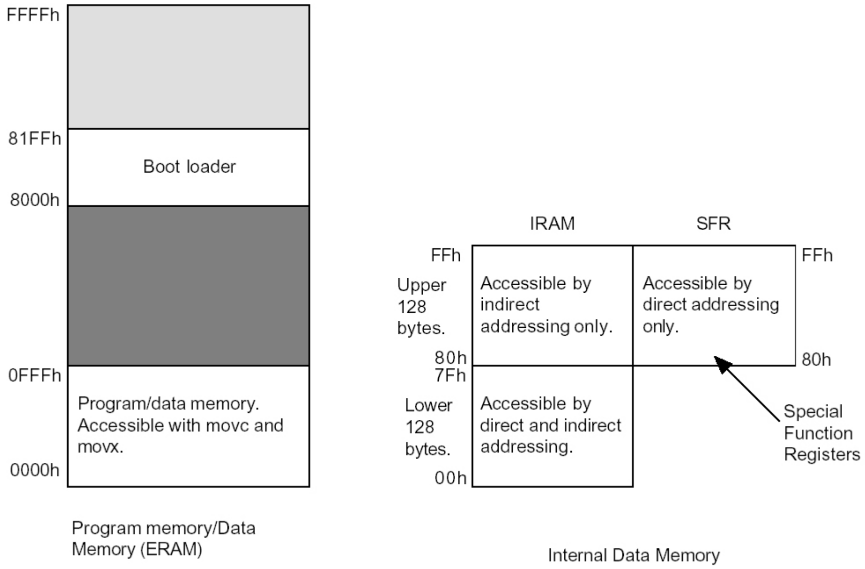
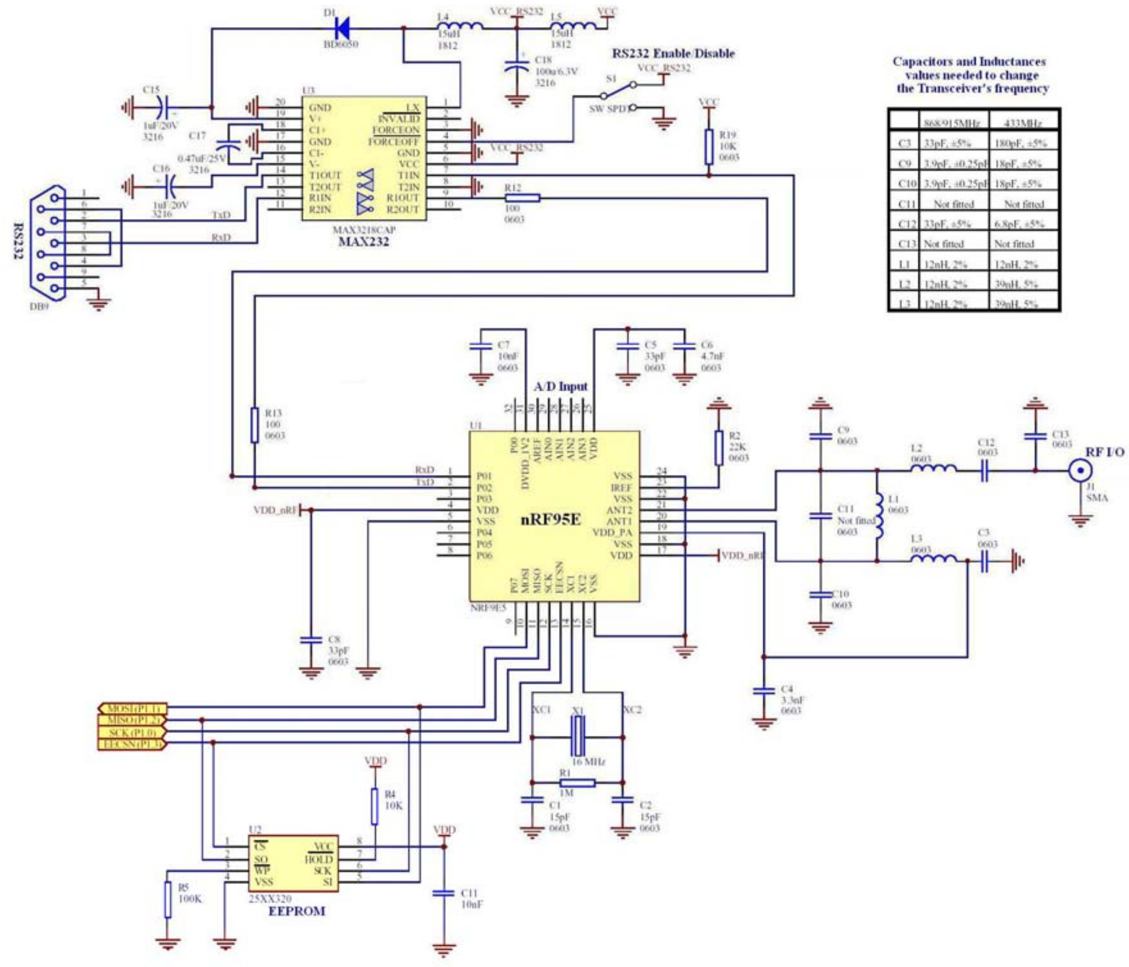

FFFFh
81FFh
Boot loader
8000h
0FFFh
Program/data memory. Accessible with movc and movx.
0000h
Program memory/Data Memory (ERAM)
IRAM
SFR
FFh
Upper 128 bytes.
Accessible by indirect addressing only.
Accessible by direct addressing only.
FFh
80h
7Fh
80h
Lower 128 bytes.
Accessible by direct and indirect addressing.
00h
Special Function Registers
Internal Data Memory

Figure 4.3.2: Memory Map of DW8051

There are 4 kB of program memory available for user programs located at the bottom of the address space as shown in Figure 4.2, which are not used in this application. The microcontroller has a 256-byte data RAM. In spite of the 512 bytes of ROM that contain a bootstrap loader (executed automatically after power is switched on) the user program is loaded into 4 kB of RAM from an external serial EEPROM by the bootstrap loader that is described in the following section.

4.3.4 Boot EEPROM/Flash

The default boot loader expects to load the user's program code from an external 25320 EEPROM. A serial programmable interface (SPI) loads the user program and once the boot process is finished, the Multiple Output Single Input (MOSI) pin P1.0, the Multiple Input Single Output (MISO) pin P1.2, and the SCK pin P1.1, are used for handling the EEPROM IC access (P1.1 and P1.2) as is shown in Figure 4.3.3.



Capacitors and Inductances values needed to change the Transceiver's frequency

	868/915MHz	433MHz
C3	33pF, ±5%	180pF, ±5%
C9	3.9pF, ±0.25pF	18pF, ±5%
C10	3.9pF, ±0.25pF	18pF, ±5%
C11	Not fitted	Not fitted
C12	33pF, ±5%	6.8pF, ±5%
C13	Not fitted	Not fitted
L1	12nH, 2%	12nH, 2%
L2	12nH, 2%	39nH, 5%
L3	12nH, 2%	39nH, 5%

Figure 4.3.3: Max 232, EEPROM IC, nRF9E5 IC and Antenna.

4.3.5 The Serial Interface

The two serial port pins for reception P0.1 (RX) and for transmission P0.2 (TX) establish the serial, full duplex, communication through a MAX 232 IC. (Figure 4.3.3). The Max 232 IC converts the TTL signal levels to RS232 levels for serial transmission or reception and it is capable of shutting down and tri-stating the outputs if a RS232 plug is not connected.

4.3.6 XTAL Oscillator

The transceiver and its embedded microcontroller use the same oscillator crystal, which runs at 16 MHz (Figure 4.3.3).

4.3.7 Antenna Output

The antenna output configuration is very important, because the output power must be used efficiently in order to attain the desired system range. A method used in radio frequency design to obtain the maximum power transfer is called “impedance matching”. Impedance matching has been used to design the evaluation kit. This method consists in matching together the antenna impedance to the transmission line impedance, and the transmission line impedance to the output impedance of the transmitter (Carr, 2001, p. 331)[24]. The Nordic evaluation kit includes a 50 Ω impedance differential (aerial) antenna matching a 50 Ω equipment impedance[25].

4.3.8 RF Data Transfer

The Universal Asynchronous Receiver Transmitter (UART) is the method used to implement software routines that transfer RF data from transceiver to transceiver[26].

[24] Carr, J.J. (2001). *Secrets of RF Circuit Design* (3rd ed.). *New York:* Mc Graw Hill.

[25] Nordic Semiconductors (2004). *Product Specification nRF9E5 EVBOARD Rev 1.0* Retrieved October 31, 2006 from: http://www.nordicsemino/

[26] Nordic Semiconductors (2002). *nRF Radio protocol guidelines nAN400-07 Revision 1.2*, Retrieved October 31, 2006 from: http://wwwnordicsemi.no/

The UART bytes form a limited payload of 32 bytes per packet being the packet format as shown in Figure 4.3.4. The packet uses a preamble signal to alert the transceiver of a new packet’s reception.

Preamble	Address	Payload	Checksum

Figure 4.3.4: UART RF Packet Format

The address identifies the device’s address. The checksum confirms the packet integrity by calculating the length of the address and payload bytes.

During the transmission-reception process, the transceiver performs the following tasks:

On the transmitter side, the radio is automatically powered up. A data packet is completed (preamble added, checksum calculated) and transmission begins at 100 kbps, GFSK modulation. The Data Ready (DR) register is set high when transmission is completed.

Then when the register called AUTO_RETRAN is set high, the transceiver continuously retransmits the packet until TRX_CE (reception flag) is set low.

On the reception side, after the receiver validates a good packet, it removes the preamble, address, checksum bytes and receives the entire payload. It then gets ready to receive more packets.

4.3.9 Output Power Software Adjustments

Output power can be programmed by configuring the PA_PWR(Power) register at the values shown in Table 4.3.3 (Nordic, 2004).

Table 4.3.3 : Configuring the RF Output Power for the nRF905

Power Settings	RF output power	DC current consumption
00	-10 dBm	11.0 mA
01	-2 dBm	14.0 mA
10	6 dBm	20.0 mA
11	10 dBm	30.0 mA

Although the default power value of the transceiver is –10 dBm, the output power has been configured to its maximum level of 10 dBm resulting in an equivalent output power of 10 mW ($10 \log_{10}(10mW) = 10$ *dBm).*

4.4 Software Application Program

The application program consists of two different codes: one for the transmission board and the second one for the reception board.

4.4.1 The nRF9E5 Configuration

The UART is configured with an interface called the “Serial Programmable Interface” (SPI). Registers SPI_DATA, SPI_CTRL and SPICLK control the SPI hardware. The SPI is made up of five internal registers (Figure 4.4.1) (Nordic, 2004) and an instruction set to indicate the different tasks to be undertaken. The data lines are MISO, MOSI and SCK. Whenever CSC is low, the interface expects an instruction, since the SPI is controlled with an specific instruction set. The Status register contains the Data Ready (DR) bits and the Address Match (AM) bit for data packets received or sent. The RF register contains the transceiver information concerning the frequency and the output power, among others.

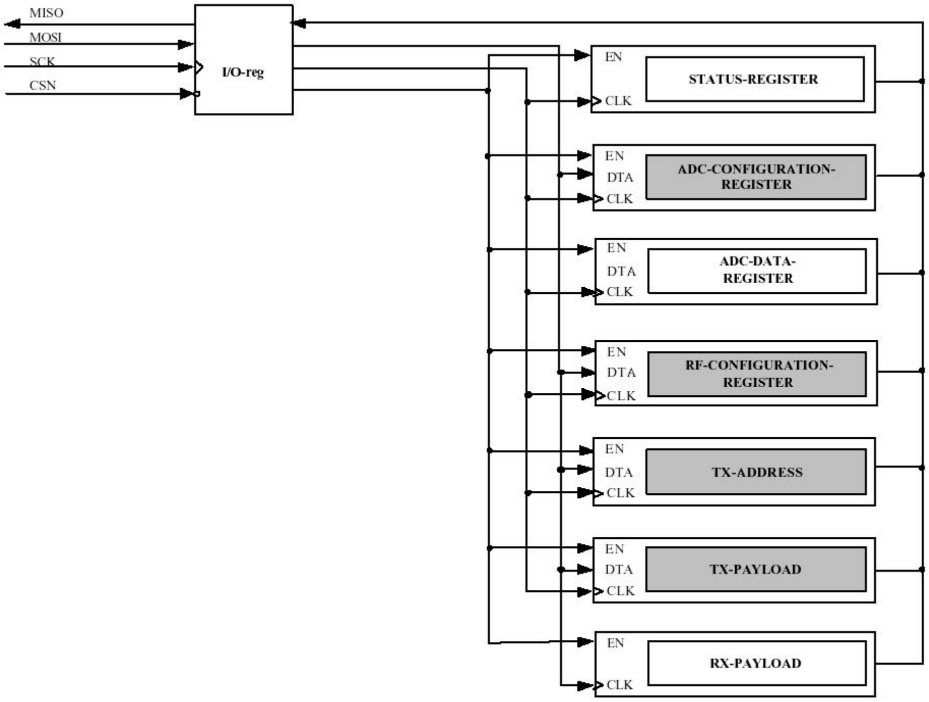


Figure 4.4.1: SPI Register and its Internal Registers

The TX-Address register retains the address of the target device. Finally, TX-Payload and RX-Payload registers, contain the payload information (to be transmitted or to be sent). It is important to mention that the limit of the TX-Payload and RX-Payload registers is 32 bytes for each packet. The ADC register is not used in this application.

4.4.2 Initialisation Routines

By using the function defined as SPI (Appendix C.3 Radio Frequency Stage Code), each byte in the function is moved and sent to the SPI data register.

The main tasks carried out during the initialization routines are:

- The SPI Clock is set at 8 MHz with the SPICLK register.

- By writing to the SPI_CTRL register, the SPI register and the nRF9E5 radio transceiver are internally connected.
- Different parameters are set such as: CPU clock frequency, crystal oscillator frequency, CRC enabled and CRC check bit enabled by writing to byte 9 in the configuration register.
- Default values are set in the payload registers by writing to bytes 3 and 4 (RX and TX payload registers) in the configuration register.
- Parameters such as: channel frequency and synthetiser's PLL frequency is set at 868 MHz by writing to the CHANNEL_CONFIG register. The output power is also set at 10 dBm (its maximum value) with this register.

On the other hand, the main UART initialisation routines are (Appendix C.3 Radio Frequency Stage Code):

- The baud rate is set with register TH1 at 9600 bps and the CPU speed at 16 MHz (the crystal oscillator speed).
- Register SCON defines the serial port operation mode, in this case mode 1, that provides an asynchronous full duplex communication mode, using 10 bits: one start bit, eight data bits and one stop bit.
- Registers TMOD and TR1 enable timer number 1.
- P0_ALT register sets pins P0.1 and P0.2 as communication ports.
- P0_DIR register assigns pin P0.1 as in input port (the RX reception port).

4.4.3 Reception Code

The code associated to the reception of data packets is divided in two different functions - ReceivePacket() and ReceiveChar(). ReceivePacket() is a sub function of ReceiveChar() (Appendix C.3 Radio Frequency Stage Code).

Figure 4.4.2 illustrates the Receive Packet Function State Machine.

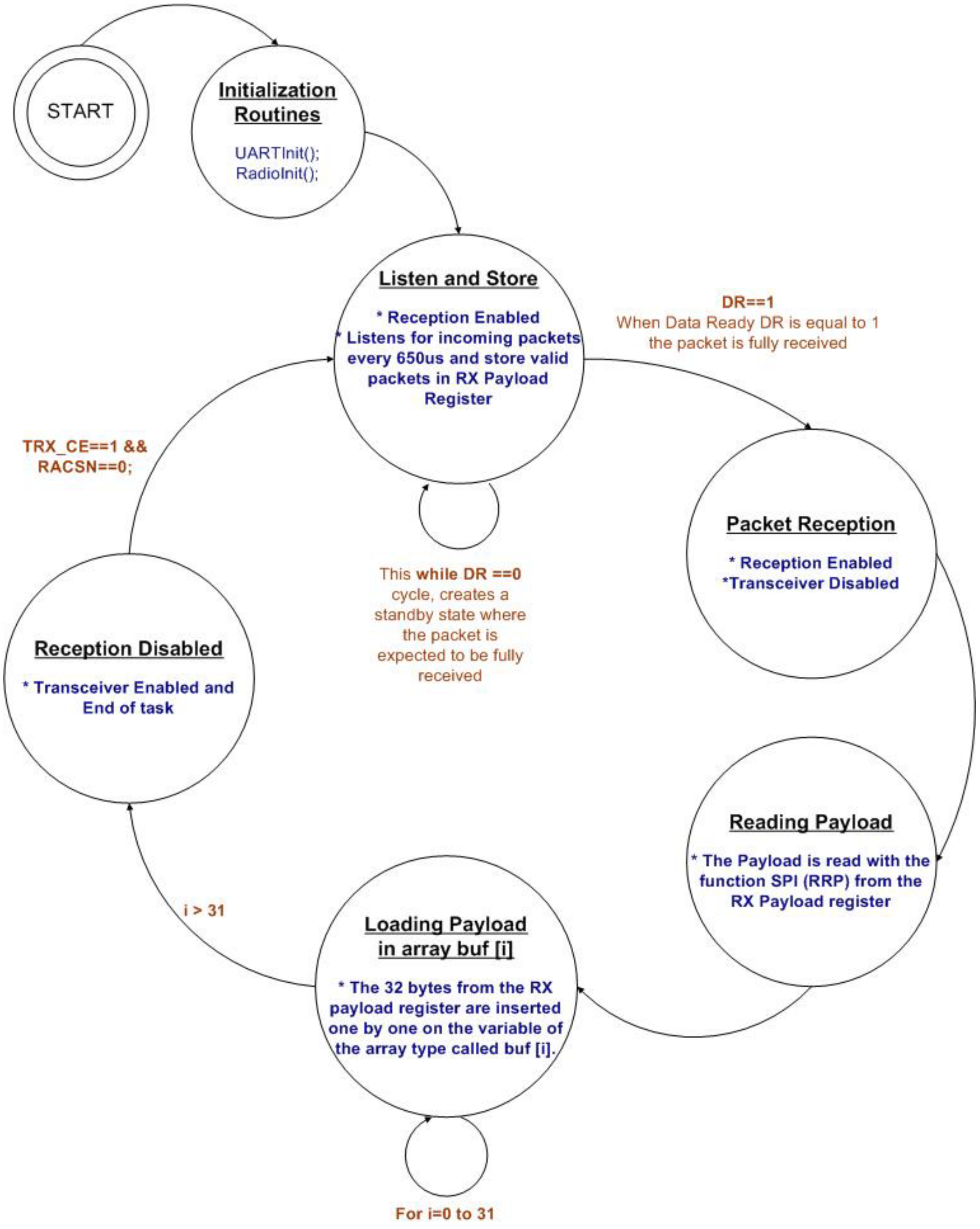

START
Initialization Routines
UARTInit();
RadioInit();
Listen and Store
* Reception Enabled
* Listens for incoming packets every 650us and store valid packets in RX Payload Register
DR==1
When Data Ready DR is equal to 1 the packet is fully received
Packet Reception
* Reception Enabled
*Transceiver Disabled
This while DR ==0 cycle, creates a standby state where the packet is expected to be fully received
Reading Payload
* The Payload is read with the function SPI (RRP) from the RX Payload register
Loading Payload in array buf [i]
* The 32 bytes from the RX payload register are inserted one by one on the variable of the array type called buf [i].
For i=0 to 31
i > 31
Reception Disabled
* Transceiver Enabled and End of task
TRX_CE==1 &&
RACSN==0;

Figure 4.4.2: Receive Packet Function State Machine

The ReceiveChar() function makes use of the previously analysed function ReceivePacket() (Appendix C.3 Radio Frequency Stage Code).

The code is explained with the following state machine in Figure 4.4.3. The function ReceiveChar(), is in charge of loading the Serial Buffer with the final data packet (retrieved from ReceivePacket() function) and can be recalled more than one time in order to finish sending FLC message. In other words if the received FLC data packet is bigger than 32 bytes (the size of the RX payload register), the FLC packet could be transmitted in more than one nRF9E5 packet.

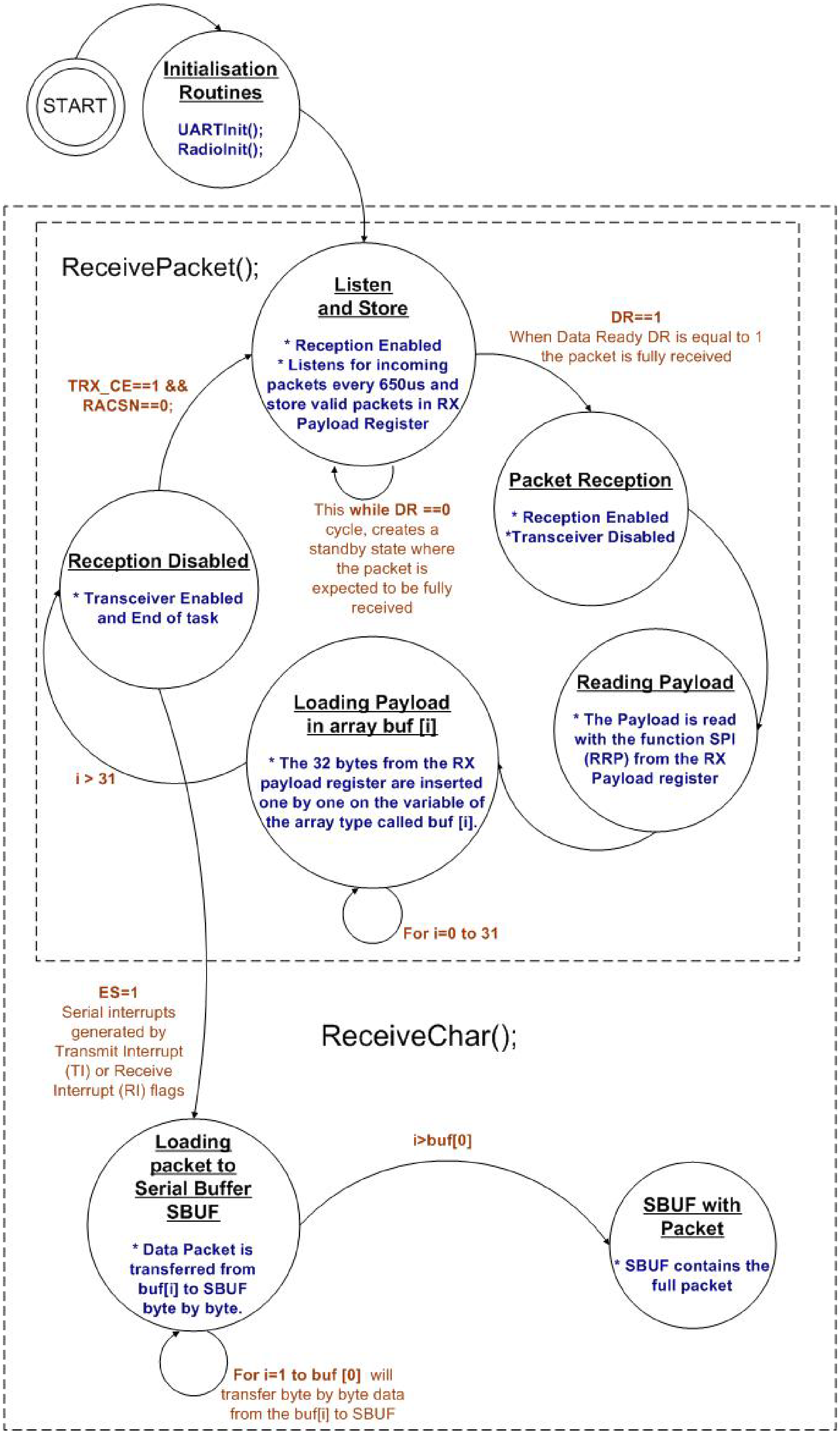

START
Initialisation Routines
UARTInit();
RadioInit();
ReceivePacket();
Listen and Store
* Reception Enabled
* Listens for incoming packets every 650us and store valid packets in RX Payload Register
DR==1
When Data Ready DR is equal to 1 the packet is fully received
Packet Reception
* Reception Enabled
*Transceiver Disabled
TRX_CE==1 &&
RACSN==0;
This while DR ==0 cycle, creates a standby state where the packet is expected to be fully received
Reception Disabled
* Transceiver Enabled and End of task
Reading Payload
* The Payload is read with the function SPI (RRP) from the RX Payload register
Loading Payload in array buf [i]
* The 32 bytes from the RX payload register are inserted one by one on the variable of the array type called buf [i].
i > 31
For i=0 to 31
ES=1
Serial interrupts generated by Transmit Interrupt (TI) or Receive Interrupt (RI) flags
ReceiveChar();
i>buf[0]
Loading packet to Serial Buffer SBUF
* Data Packet is transferred from buf[i] to SBUF byte by byte.
SBUF with Packet
* SBUF contains the full packet
For i=1 to buf [0] will transfer byte by byte data from the buf[i] to SBUF

Figure 4.4.3: ReceiveChar Function State Machine

4.4.4 Transmission Code

The transmission code is integrated by two functions TransmitChar() and GetChar() (Appendix C.3 Radio Frequency Stage Code). The main program will also be analysed in a later section.

The goal of this code is to transmit a data packet via the radio frequency link. The packet is first written from the serial buffer to the transceiver's TX payload register and then it is sent though the radio frequency link as is shown in the following state machine of Figure 4.4.4.

Also GetChar() is used in the main program to recover each byte present at the serial buffer register (SBUF).

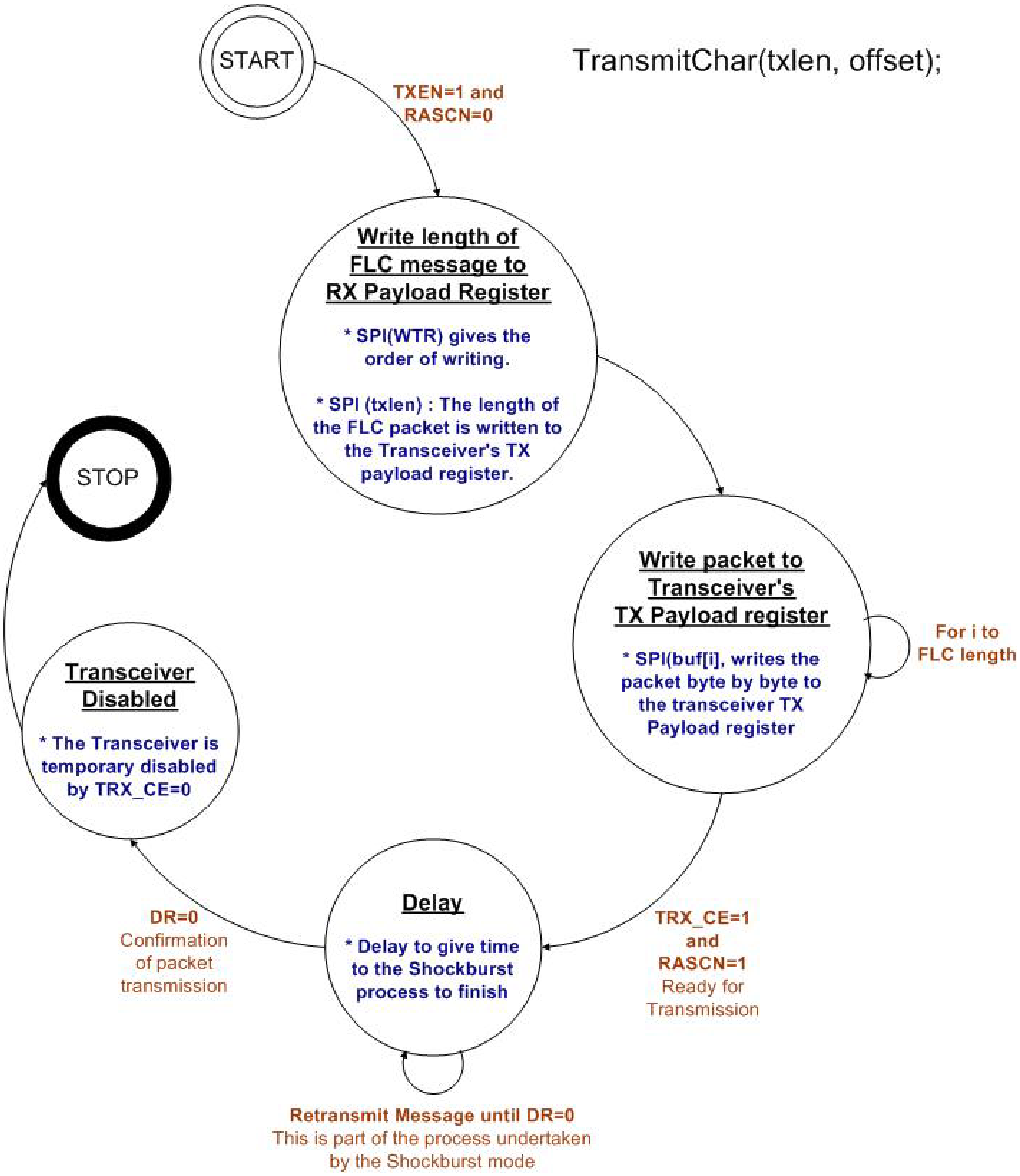

START
TransmitChar(txlen, offset);
TXEN=1 and RASCN=0
Write length of FLC message to RX Payload Register
* SPI(WTR) gives the order of writing.
* SPI (txlen) : The length of the FLC packet is written to the Transceiver's TX payload register.
STOP
Write packet to Transceiver's TX Payload register
* SPI(buf[i], writes the packet byte by byte to the transceiver TX Payload register
For i to FLC length
Transceiver Disabled
* The Transceiver is temporary disabled by TRX_CE=0
Delay
* Delay to give time to the Shockburst process to finish
TRX_CE=1 and RASCN=1
Ready for Transmission
DR=0
Confirmation of packet transmission
Retransmit Message until DR=0
This is part of the process undertaken by the Shockburst mode

Figure 4.4.4: TransmitChar Function State Machine

The main program (Appendix C.3 Radio Frequency Stage Code) undertakes the task of integrating the different functions and transmitting a wireless FLC data packet that has been detected in the serial port. The limit for transmitting one transceiver packet is 32 bytes, however if the FLC packet is bigger than that size, the first 32 bytes would be sent and the rest of the packet would be transmitted in a second packet.

The process of the transmission’s main program is better represented in the following flow diagram (Figure 4.4.5).

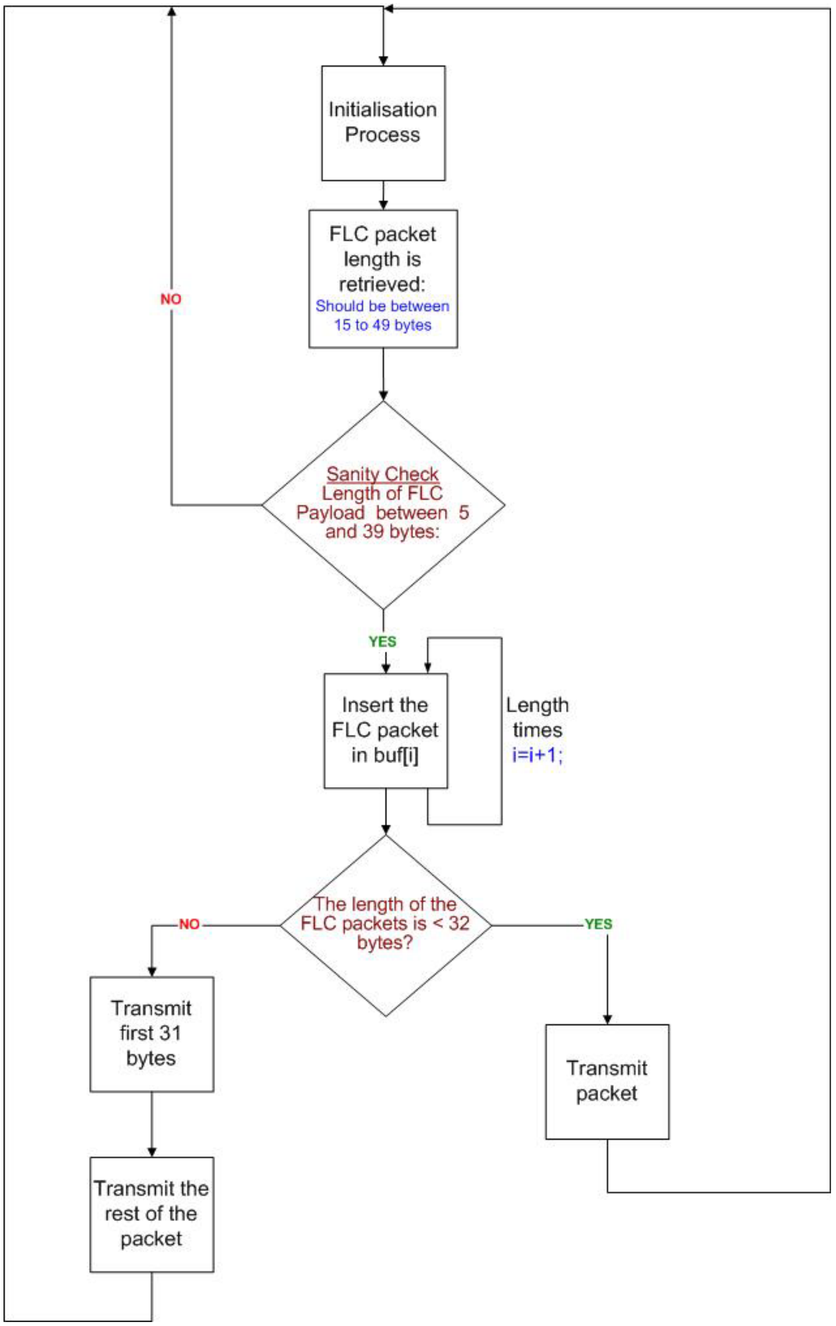

Initialisation Process
FLC packet length is retrieved:
Should be between 15 to 49 bytes
Sanity Check
Length of FLC Payload between 5 and 39 bytes:
NO
YES
Insert the FLC packet in buf[i]
Length times
i=i+1;
The length of the FLC packets is < 32 bytes?
NO
YES
Transmit first 31 bytes
Transmit the rest of the packet
Transmit packet

Figure 4.4.5: Main program for the Transmission stage

4.5 Prototype Integration

The final prototype was integrated with the following sections:

- Enclosure (Figure 4.5.1)
- Power: This section included a 14 V solar cell to load a 12 V Battery that supplied the needed power for the three boards included in this final prototype. (Figure 4.5.1)

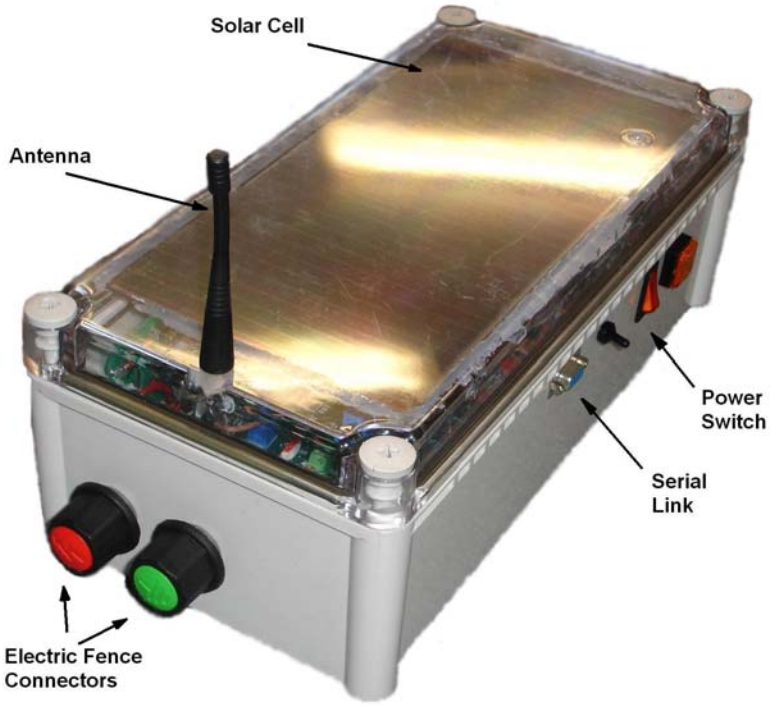
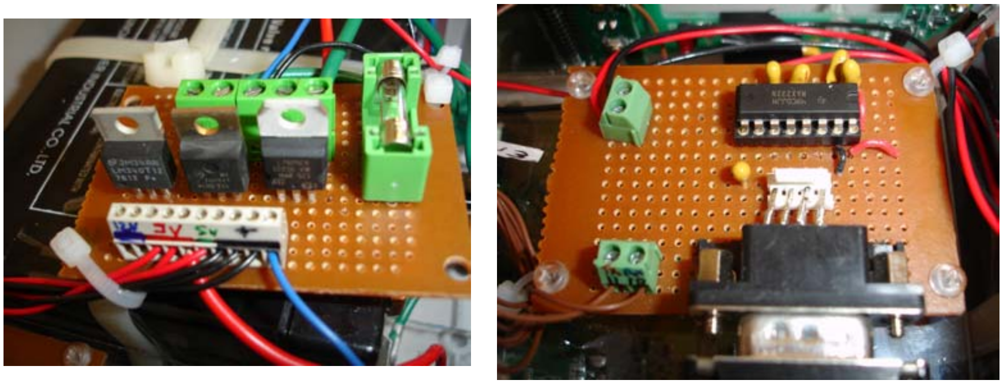


Figure 4.5.1: Prototype's Enclosure and Some External Components

- Voltage regulators to supply 12 V and 5 V (Figure 4.5.2)
- A MAX232 board to give the desired TTL voltage levels for serial communication. (Figure 4.5.2)



Voltage regulators

MAX 232

Figure 4.5.2: Voltage Regulators and MAX 232 Circuits

- The three electronic boards (Figure 4.5.3) were installed behind the batteries: the energizer’s logic board, the SMART PAC board and the Nordic evaluation kit board.

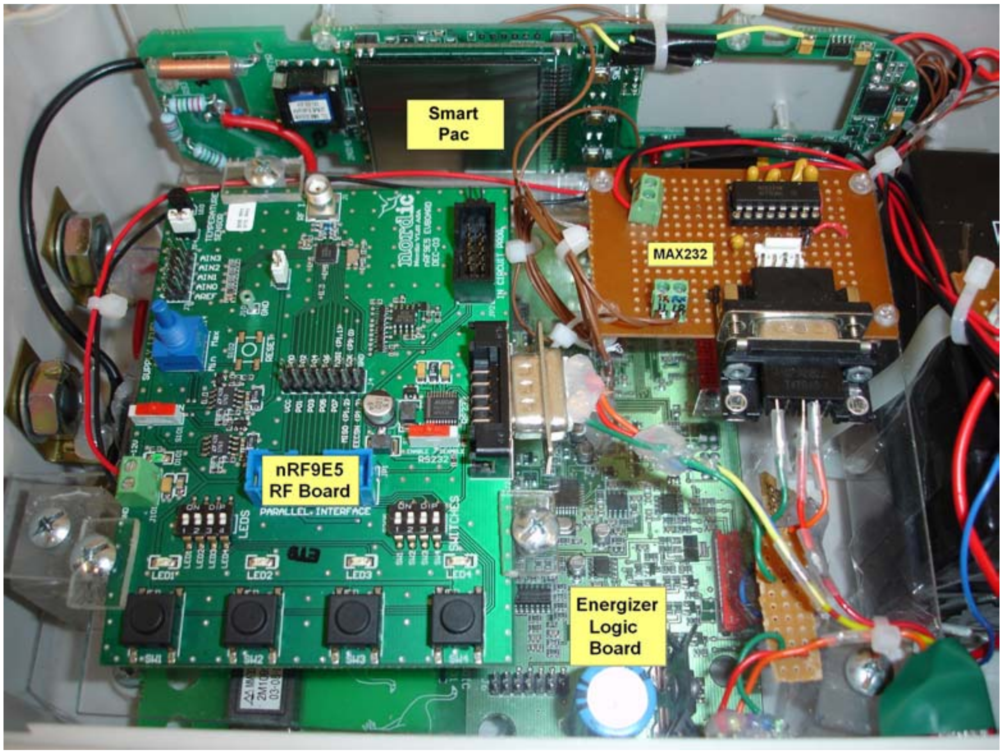


Figure 4.5.3: Electronic Boards Mounted in Enclosure

The final network had the following topology (Figure 4.5.4) as expected.

When a FLC message is generated along the electric fence by an energizer, (Figure 4.5.4, [1]), the logic board (Figure 4.5.4, [2]) detects it and sends it to the transmission RF board (Figure 4.5.4 [3],), which transmits the message via RF to the reception RF board (Figure 4.5.4, [4]). The reception RF board receives the message and sends it via serial port to the farmer's PC board (Figure 4.5.4, [5]). The followed path was: [1], [2], [3], [4] and [5].

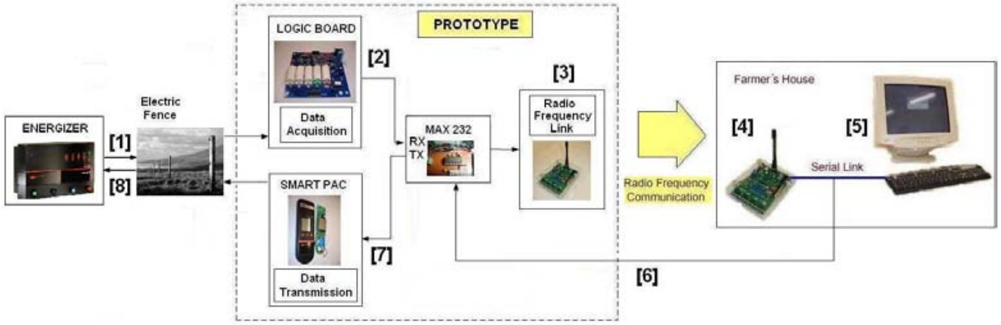


Figure 4.5.4: Final Network Topology

If the communication takes place in the opposite direction the message is generated from the farmer's PC (Figure 4.5.4, [5]) and sent via serial port (Figure 4.5.4, [6]) to the prototype's SMART PAC board (Figure 4.5.4, [7]). Then the SMART PAC board sends the data packet to the electric fence and is received by the energizer (Figure 4.5.4, [8]). The followed path was: [5], [6], [7] and [8].

4.6 Final Tests

The first tests were carried out in the laboratory, as stated in Sections 2.5.1 and 3.6.1. However the final tests were undertaken in a farm with an electric fence.

The first tests consisted of verifying the correct transmission and reception of messages using the prototype.

The transmission messages were sent from the farmer’s PC to the electric fence. The messages had the same format as the ones that were generated with a SMART PAC remote control: On, Off and Inquire messages. These messages were typed and sent to the serial port of a PC through the HyperTerminal software (Figure 4.6.1, [1]). These messages were processed by the SMART PAC electronic board and then sent to its serial output that was directly connected acquisition logic board (Figure 4.6.1, [2]). The message was then collected by the energizer’s logic board and transmitted to another serial port of the same PC where the messages were originally generated (Figure 4.6.1, [3]). Therefore, HyperTerminal displayed the transmitted messages as received ones (Figure 4.6.1, [4]).

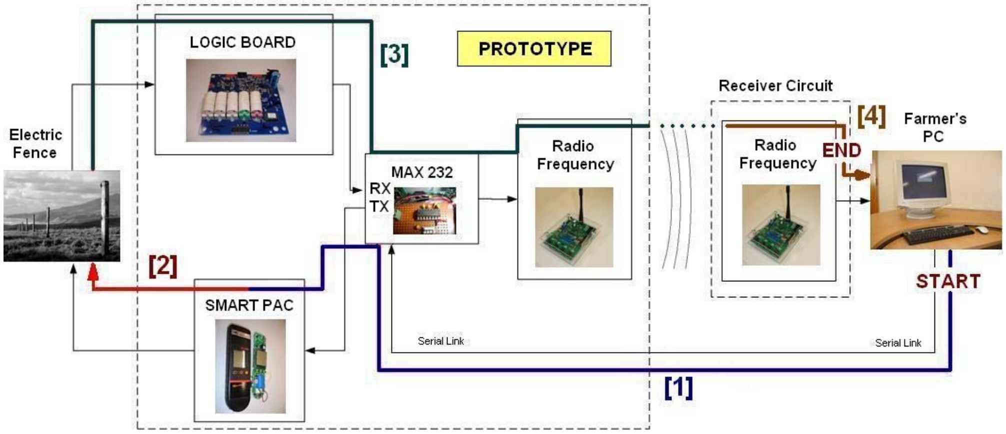
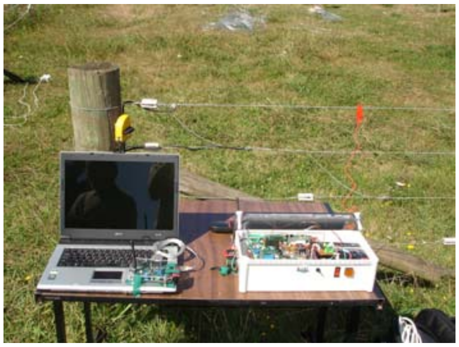


Figure 4.6.1: Sending Messages to the Electric Fence

For the tests undertaken on the farm (Figure 4.6.2), the wireless link prototype was coupled to the electric fence and the reception board was connected to a laptop PC in order to verify the reception of any data transmitted through the fence.



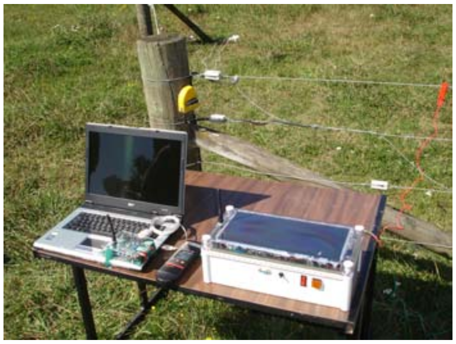


Figure 4.6.2: Tests on a Farm

The tests consisted of transmitting FLC messages to the electric fence using a SMART PAC remote control (Figure 4.6.3, [1]) and then seeing their correct reception in a laptop. Therefore the transmitted messages were collected by the wireless link prototype and then they were sent via radio frequency to the laptop (Figure 4.6.3, [2]). HyperTerminal displayed such messages on the laptop (Figure 4.6.3, [3]). Trials confirmed that the maximum transmission range was around 250 m. As stated in section 4.2.2 Early Stages Test; the expected range was around 500 m.

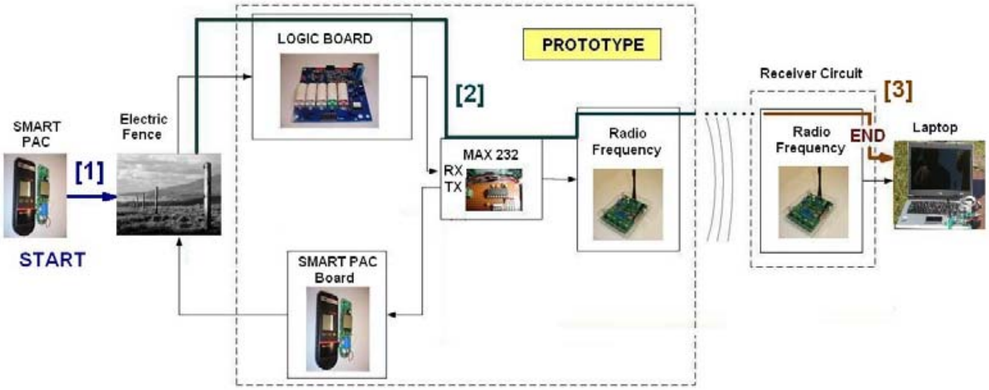


Figure 4.6.3: Sending Messages from the Fence to a Laptop

The early tests and the final tests took place in similar conditions, however, the final prototype's tests added more RF noise sources. Some of these sources could be, for example: different circuits working together, power supplies and internal ADCs working at high frequencies. These noise generators could have reduce the ACS that determines the receiver's noise immunity and therefore the RF range reached. Another variation came from the size of information that was transmitted and received. The prototype sent packets of 32 bytes at 9600 bits per second.

Future enhancements are proposed to avoid concerns about RF range:

- Increasing the RF output power by building a "purpose built" Printed Circuit Board (PCB). At 868 MHz, in New Zealand it is possible increase the power up to 1 W (Table 4.2.1 : Frequencies and Power).
- Even though an aerial antenna should be part of the network, special attention should be given to the design of the small loop

antenna, which is an integral part of the PCB. Small loop antennas designed with a narrow bandwidth suppress unwanted signals that disturb the receiver.

- The radio range could be improved by keeping some free space around the antenna, because objects nearby will form part of the tuning network and could be tuning the antenna away from the wanted frequency.[27]
- By adding two band-pass filters (built on the PCB) to the design, the ACS could be dramatically increased[28].

4.7 Summary of Results

The prototype is able to establish communication with the electric fence, as stated in Section 1.5: the farmer's PC receives wirelessly the FLC packets travelling along the fence and it can send messages to the fence using the serial link.

[27] Nordic Semiconductors (2002), *Design-in of RF circuits,* Retrieved October 31, 2006 from http://www.nordicsemi.no/

[28] Fabrikant, A.L. (1989). A method of increasing adjacent channel selectivity in receivers, *Telecommunications and radio engineering, 44,* 91-92.

Chapter 5.
Results and Conclusion

5.1 Global results

This brief chapter compiles the overall results of the project, the problems faced as well as the possible improvements that can be carried out in the future for the development of a product.

5.1.1 Results in Acquisition and Transmission Stages

A detailed analysis of Gallagher’s electronic boards was essential to define the design approach to suit the project’s requirements. Both electronic boards were reprogrammed and most of their hardware components were unchanged.

In chapter one, the initial requirements defined that products such as the energizer’s electronic board should be considered as a starting platform to develop the prototype.

Using actual products, as a starting point of this project was obviously very useful. However, there are disadvantages. For example, needing to interpret the original DSP’s software code and partially modify it. Understanding the original software code was complicated, however it also allowed DSP functions to be identified, which led to a better implementation of a prototype. If this code is improved in the future, it could be possible to pack together the two stages (data acquisition and data transmission) into one, by using a single DSP.

5.1.2 Results in the Radio Frequency Link Stage

Two evaluation boards were used as radio frequency links. Two different programs were loaded on each board: one for transmitting and the other for receiving. The initial requirements defined in chapter one were fulfilled in the following subjects:

- The frequencies for the wireless devices are configurable 433, 868 and 915 MHz.
- The farmer's PC can: A) Receive data packets that travel along the electric fence and B) Transmit data packets to the electric fence.
- The output power is also configurable by modifying the software.
- Known technologies such as Nordic Semiconductor, ATMEL and Chipcon were considered, however the chosen brand was Nordic Semiconductors.
- The radio frequency devices were able to establish communication through walls and other obstacles.

5.1.3 Range and Power Limitations

Range should have been greater than that achieved by the final tests. A reduction of the output power or a lack of reception sensitivity could have been the reason for this difficulty.

5.1.4 Software Limitations

The RF programmed code allows FLC messages to be broadcast from the electric fence to the farmer's PC. Although the prototype's hardware fully supports wireless communication in both directions, for testing and measurement purposes, PC to fence communication was carried out via serial cable. This limitation can be overcome by modification of the current software.

5.2 Potential Enhancements

The next sections highlight some of the most relevant proposals to improve the prototype.

5.2.1 Hardware

The design and implementation of a “purpose built” PCB as stated in chapter 4.6 (Final Tests), would radically enhance RF network performance. This PCB design should consider removing the actual noise sources and increasing the ACS.

5.2.2 Software

The original DSP code of the energizer’s logic board and the SMART PAC remote control could be reduced to a single compact code module for acquiring and transmitting data. The use of a single DSP would be a further step in the development.

For a future product development the RF boards should be reprogrammed to achieve communication in both directions by implementing interrupt routines that coordinate reception and transmission activities. Register “tx_address” from SPI could be used to acknowledge the ID of different devices transmitting RF messages.

5.3 Conclusions

The successful design and implementation of a prototype wireless link to monitor an electric fence consisted of an integration of several off-the-shelf technologies. This resulted in most of the goals being implemented, however a purpose built PCB design, could be developed in the future that would be better able to meet all the goals.

Nonetheless, this is a good first step in the research of a marketable product that will be of assistance to the farmer of the future.

Appendix A: References

Chapter 1. Background

[1] Gallagher Group LTD (2005). *How to?, How does an electric fence work?.* Retrieved March 6, 2006 from: http://www.gallagher.co.nz/pf.downloads.aspx

[2] Gallagher Group LTD (2005). *Gallagher Power Fence System Manual* (12th ed.). Hamilton: Gallagher Group.

[3] Gallagher Group LTD (2005). *Energizer's Overview.* Retrieved March 6, 2006 from: http://www.gallagher.co.nz/energizer.overview.aspx

[4] Gallagher Group LTD (2005). *Gallagher Power Fence System Manual* (12th ed.). Hamilton: Gallagher Group.

[5] Stafix Fence Alert (2005). *Fence Monitors*. Retrieved March 6, 2006 from: www.true-test.com/pel_new2

[6] Pakton (2005). *PTE0480 Electric Fence Monitor.* Retrieved March 6, 2006 from: http://www.pakton.com.au/pte0480.html

Chapter 2. Data Acquisition

[7] Parker, S. & Roberts, V. (2003). *Fence Line Communications for GGL Fence Management Systems V.0.3 Protocol Specification.* Hamilton: Gallagher Group.

[8] Gallagher Group LTD (2005). *MX7500 230V AC Mains Fence Energizers Service Manual* (1st ver.). Hamilton: Gallagher Group.

[9] Gibson, J. (1989). *Principles of Digital and Analogue Communications*. New York: Macmillan Publishing Company.

[10] Long, M. (2002). *Murray Longs Fence Coupling, Technical Summary*. Hamilton NZ, Gallagher Group.

[11] Texas Instruments Incorporated (2005). *TMS320LF2401A DSP Controller Datasheet*. Retrieved March 15, 2006 from: http://www.ti.com

[12] Body, N. (2002). *SMART PAC FSK Modem Documentation, Signal Flow Graph*. Hamilton: Gallagher Group.

[13] Body, N. (2005). *Functional Flow Diagram SMART PAC – Remote and Fault Finder, High-Level Design* (3rd rev.). Hamilton NZ, Gallagher Group.

Chapter 3. Data Transmission

[14] Guru, B.S. & Hiziroglu, H.R. (1998). *Electromagnetic Field Theory Fundamentals*. Boston: PWS Publishing Company.

[15] Staelin, D.H., Mortgenthaler, A.W. & Kong, J. A. (1994). *Electromagnetic Waves.* New Jersey: Prentice Hall.

Chapter 4. Radio Frequency Link

[16] Karlsen, F. (2000). *Guidelines to low cost wireless system design*. Retrieved October 31, 2006 from: http://www.nordicsemi.no/

[17] Carr, J.J. (2001). *Secrets of RF Circuit Design* (3rd ed.). New York: Mc Graw Hill.

[18] NZ Gazette (2003). *Notice: go3206, Radio Communications Regulations*. Retrieved April 17, 2006 from: http://www.secnet.co.nz/

[19] ERO (2006). *The European Table of Frequency Allocations and Utilisations, Report 25*. Retrieved April 17, 2006 from: http://www.ero.dk/

[20] Losee, F.A. (2005). *RF Systems, components, and circuits handbook,* (2nd ed.). Massachusetts: Artech House.

[21] ETSI (1999). *Radio Transmissions and Reception, UMTS XX.06 Version 1.0.0.* Retrieved April 17, 2006 from: http://www.3gpp.org/ftp/tsg_ran/WG4_Radio/TSGR4_01/Docs/Others/P-99-097.doc

[22] Lenk, J.D. (1999). *Optimizing Wireless / RF Circuits*. New York: McGraw Hill.

[23] Nordic Semiconductors (2004). *Product Specification nRF9E5* (1.1 rev.). Retrieved October 31, 2006 from: http://www.nordicsemi.no/

[24] Proakis, J.G. (2001). *Digital Communications* (4th ed.). Boston: McGraw Hill.

[25] Carr, J.J. (2001). *Secrets of RF Circuit Design* (3rd ed.). New York: Mc Graw Hill.

[26] Nordic Semiconductors (2004). *Product Specification nRF9E5 EVBOARD Rev 1.0*. Retrieved October 31, 2006 from: http://www.nordicsemi.no/

[27] Nordic Semiconductors (2002). *nRF Radio protocol guidelines nAN400-07 Revision 1.2*. Retrieved October 31, 2006 from: http://www.nordicsemi.no/

[28] Nordic Semiconductors (2002). *Design-in of RF circuits.* Retrieved October 31, 2006 from: http://www.nordicsemi.no/

[29] Fabrikant, A.L. (1989). A method of increasing adjacent channel selectivity in receivers, *Telecommunications and radio engineering, 44,* 91-92.

Appendix B: Data Sheets

B.1 Mitsubishi Microcontroller and Texas Instrument DSP

Mitsubishi microcontroller (Figure B.1.1)

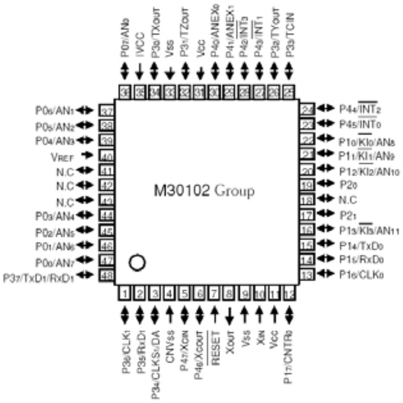
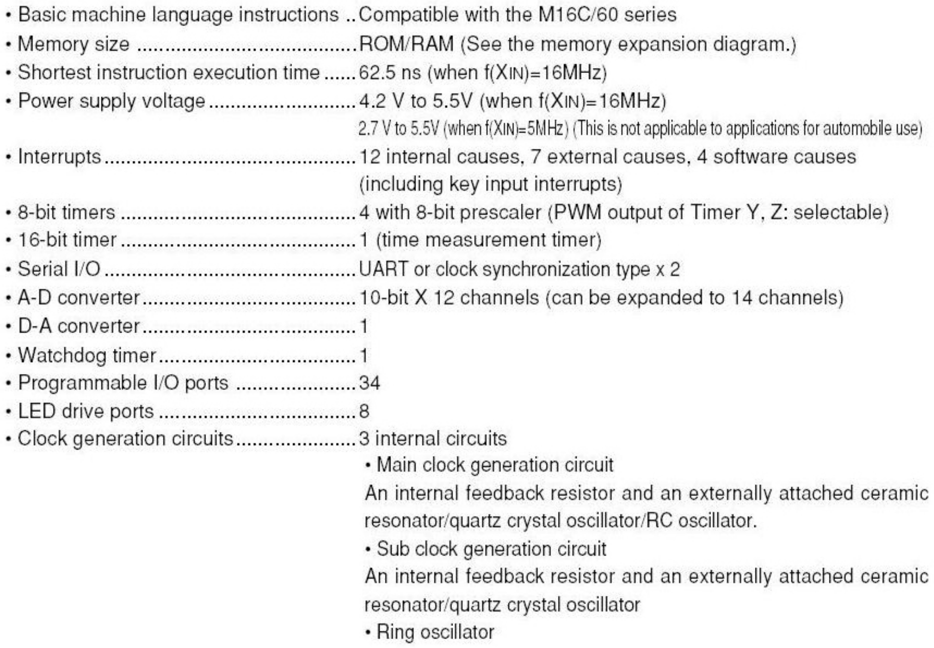


Figure B.1.1 : Mitsubishi M30100/M30102 Group Microcontroller. [29]

Features:



- Basic machine language instructions ..Compatible with the M16C/60 series
- Memory size ....ROM/RAM (See the memory expansion diagram.)
- Shortest instruction execution time ......62.5 ns (when $f(X_{IN})=16MHz$)
- Power supply voltage ....4.2 V to 5.5V (when $f(X_{IN})=16MHz$)
  2.7 V to 5.5V (when $f(X_{IN})=5MHz$) (This is not applicable to applications for automobile use)
- Interrupts ....12 internal causes, 7 external causes, 4 software causes
  (including key input interrupts)
- 8-bit timers ....4 with 8-bit prescaler (PWM output of Timer Y, Z: selectable)
- 16-bit timer ....1 (time measurement timer)
- Serial I/O ....UART or clock synchronization type x 2
- A-D converter......................................10-bit X 12 channels (can be expanded to 14 channels)
- D-A converter......................................1
- Watchdog timer...................................1
- Programmable I/O ports 34
- LED drive ports 8
- Clock generation circuits......................3 internal circuits
  - Main clock generation circuit
    An internal feedback resistor and an externally attached ceramic resonator/quartz crystal oscillator/RC oscillator.
  - Sub clock generation circuit
    An internal feedback resistor and an externally attached ceramic resonator/quartz crystal oscillator
  - Ring oscillator

[29] Renesas Technology Corporation (2003), *Tentative Specifications* (rev. e1). Retrieved March 15, 2006 from: http://www.mitsubishichips.com

Logic Board DSP (Figure B.1.2)

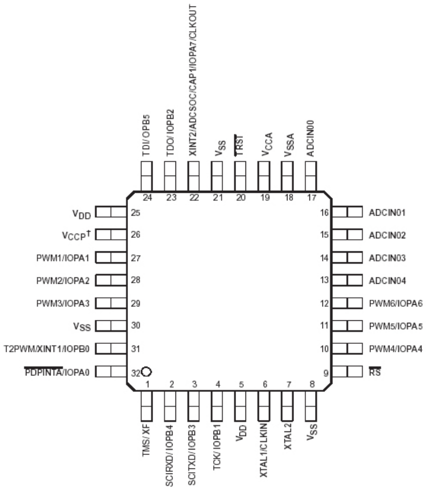
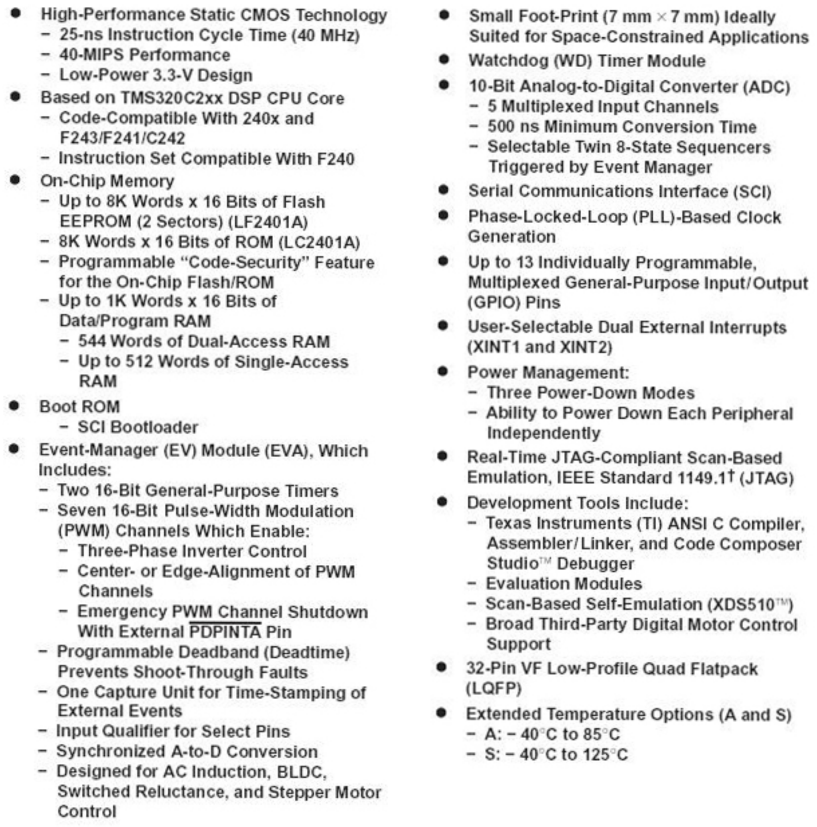


Figure B.1.2 : Texas Instruments DSP TMS320LF2401A[30].

Features:



- High-Performance Static CMOS Technology
  - 25-ns Instruction Cycle Time (40 MHz)
  - 40-MIPS Performance
  - Low-Power 3.3-V Design
- Based on TMS320C2xx DSP CPU Core
  - Code-Compatible With 240x and F243/F241/C242
  - Instruction Set Compatible With F240
- On-Chip Memory
  - Up to 8K Words x 16 Bits of Flash EEPROM (2 Sectors) (LF2401A)
  - 8K Words x 16 Bits of ROM (LC2401A)
  - Programmable "Code-Security" Feature for the On-Chip Flash/ROM
  - Up to 1K Words x 16 Bits of Data/Program RAM
    - 544 Words of Dual-Access RAM
    - Up to 512 Words of Single-Access RAM
- Boot ROM
  - SCI Bootloader
- Event-Manager (EV) Module (EVA), Which Includes:
  - Two 16-Bit General-Purpose Timers
  - Seven 16-Bit Pulse-Width Modulation (PWM) Channels Which Enable:
    - Three-Phase Inverter Control
    - Center- or Edge-Alignment of PWM Channels
    - Emergency PWM Channel Shutdown With External $\overline{\text{PDPINTA}}$ Pin
  - Programmable Deadband (Deadtime) Prevents Shoot-Through Faults
  - One Capture Unit for Time-Stamping of External Events
  - Input Qualifier for Select Pins
  - Synchronized A-to-D Conversion
  - Designed for AC Induction, BLDC, Switched Reluctance, and Stepper Motor Control
- Small Foot-Print (7 mm × 7 mm) Ideally Suited for Space-Constrained Applications
- Watchdog (WD) Timer Module
- 10-Bit Analog-to-Digital Converter (ADC)
  - 5 Multiplexed Input Channels
  - 500 ns Minimum Conversion Time
  - Selectable Twin 8-State Sequencers Triggered by Event Manager
- Serial Communications Interface (SCI)
- Phase-Locked-Loop (PLL)-Based Clock Generation
- Up to 13 Individually Programmable, Multiplexed General-Purpose Input/Output (GPIO) Pins
- User-Selectable Dual External Interrupts (XINT1 and XINT2)
- Power Management:
  - Three Power-Down Modes
  - Ability to Power Down Each Peripheral Independently
- Real-Time JTAG-Compliant Scan-Based Emulation, IEEE Standard 1149.1† (JTAG)
- Development Tools Include:
  - Texas Instruments (TI) ANSI C Compiler, Assembler/Linker, and Code Composer Studio™ Debugger
  - Evaluation Modules
  - Scan-Based Self-Emulation (XDS510™)
  - Broad Third-Party Digital Motor Control Support
- 32-Pin VF Low-Profile Quad Flatpack (LQFP)
- Extended Temperature Options (A and S)
  - A: − 40°C to 85°C
  - S: − 40°C to 125°C

[30] Texas Instruments (2005). *DSP Controllers Data Sheet*. Retrieved March 15, 2006 from: www.ti.com

The SMART PAC DSP (Figure B.1.3)

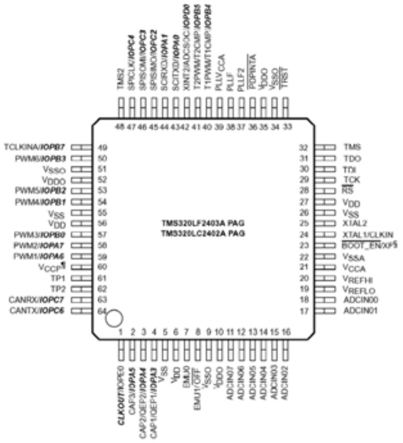
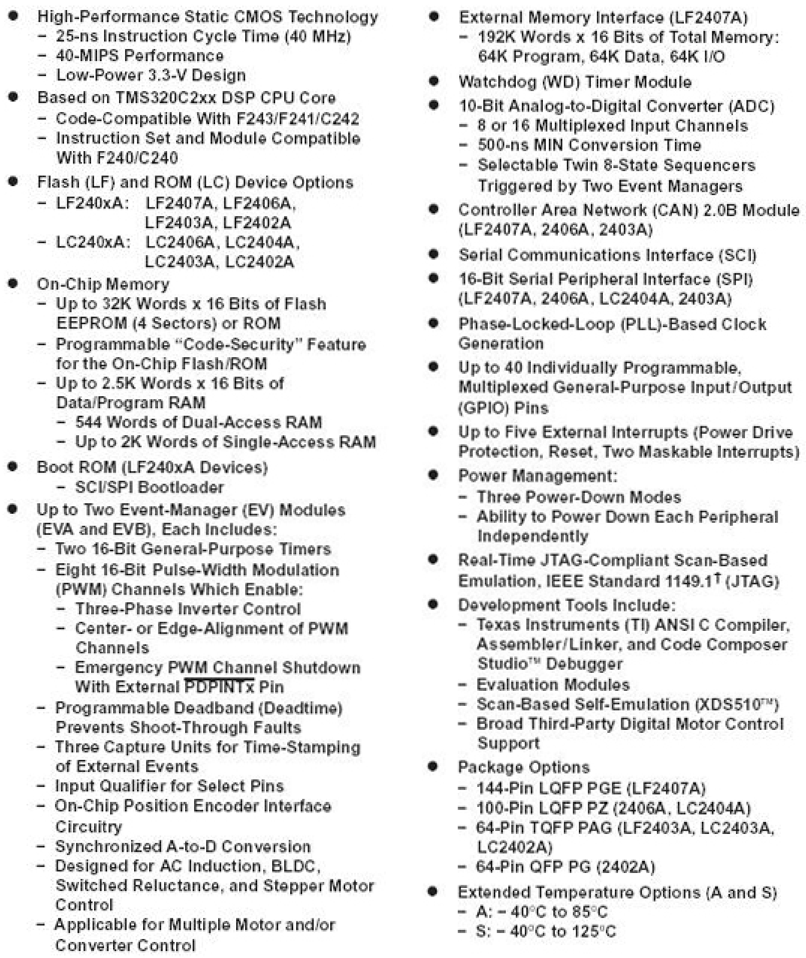


Figure B.1.3 : SMART PAC DSP TMS320LF2403A [31]

Features:



- High-Performance Static CMOS Technology
  - 25-ns Instruction Cycle Time (40 MHz)
  - 40-MIPS Performance
  - Low-Power 3.3-V Design
- Based on TMS320C2xx DSP CPU Core
  - Code-Compatible With F243/F241/C242
  - Instruction Set and Module Compatible With F240/C240
- Flash (LF) and ROM (LC) Device Options
  - LF240xA: LF2407A, LF2406A, LF2403A, LF2402A
  - LC240xA: LC2406A, LC2404A, LC2403A, LC2402A
- On-Chip Memory
  - Up to 32K Words x 16 Bits of Flash EEPROM (4 Sectors) or ROM
  - Programmable "Code-Security" Feature for the On-Chip Flash/ROM
  - Up to 2.5K Words x 16 Bits of Data/Program RAM
    - 544 Words of Dual-Access RAM
    - Up to 2K Words of Single-Access RAM
- Boot ROM (LF240xA Devices)
  - SCI/SPI Bootloader
- Up to Two Event-Manager (EV) Modules (EVA and EVB), Each Includes:
  - Two 16-Bit General-Purpose Timers
  - Eight 16-Bit Pulse-Width Modulation (PWM) Channels Which Enable:
    - Three-Phase Inverter Control
    - Center- or Edge-Alignment of PWM Channels
    - Emergency PWM Channel Shutdown With External PDPINTx Pin
  - Programmable Deadband (Deadtime) Prevents Shoot-Through Faults
  - Three Capture Units for Time-Stamping of External Events
  - Input Qualifier for Select Pins
  - On-Chip Position Encoder Interface Circuitry
  - Synchronized A-to-D Conversion
  - Designed for AC Induction, BLDC, Switched Reluctance, and Stepper Motor Control
  - Applicable for Multiple Motor and/or Converter Control
- External Memory Interface (LF2407A)
  - 192K Words x 16 Bits of Total Memory: 64K Program, 64K Data, 64K I/O
- Watchdog (WD) Timer Module
- 10-Bit Analog-to-Digital Converter (ADC)
  - 8 or 16 Multiplexed Input Channels
  - 500-ns MIN Conversion Time
  - Selectable Twin 8-State Sequencers Triggered by Two Event Managers
- Controller Area Network (CAN) 2.0B Module (LF2407A, 2406A, 2403A)
- Serial Communications Interface (SCI)
- 16-Bit Serial Peripheral Interface (SPI) (LF2407A, 2406A, LC2404A, 2403A)
- Phase-Locked-Loop (PLL)-Based Clock Generation
- Up to 40 Individually Programmable, Multiplexed General-Purpose Input/Output (GPIO) Pins
- Up to Five External Interrupts (Power Drive Protection, Reset, Two Maskable Interrupts)
- Power Management:
  - Three Power-Down Modes
  - Ability to Power Down Each Peripheral Independently
- Real-Time JTAG-Compliant Scan-Based Emulation, IEEE Standard 1149.1† (JTAG)
- Development Tools Include:
  - Texas Instruments (TI) ANSI C Compiler, Assembler/Linker, and Code Composer Studio™ Debugger
  - Evaluation Modules
  - Scan-Based Self-Emulation (XDS510™)
  - Broad Third-Party Digital Motor Control Support
- Package Options
  - 144-Pin LQFP PGE (LF2407A)
  - 100-Pin LQFP PZ (2406A, LC2404A)
  - 64-Pin TQFP PAG (LF2403A, LC2403A, LC2402A)
  - 64-Pin QFP PG (2402A)
- Extended Temperature Options (A and S)
  - A: – 40°C to 85°C
  - S: – 40°C to 125°C

[31] Texas Instruments (2005). *DSP Controllers Data Sheet*. Retrieved March 15, 2006 from: www.ti.com

B.2 Data Sheet for Details of the nRF9E5 Transceiver

Figure B.2.1 shows the block diagram of the nRF9E5 IC and the detail on the block diagram of nRF905 transceiver (Figure B.2.2). [32]

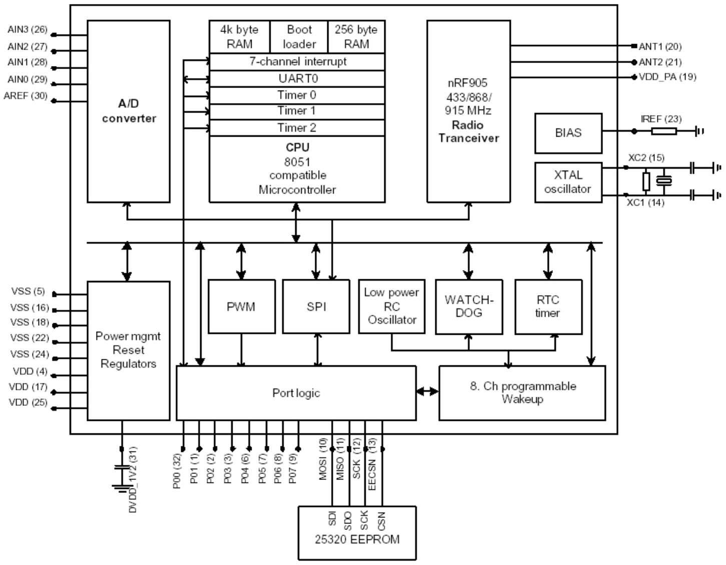


Figure B.2.1: nRF9E5 Block Diagram

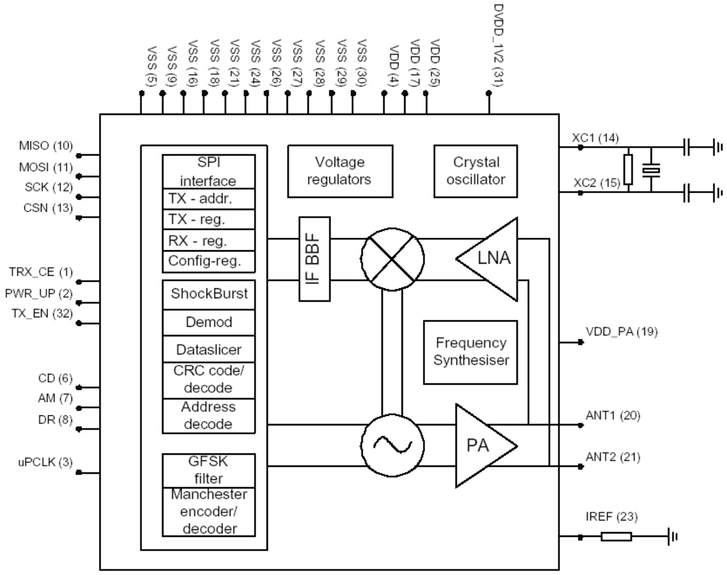


Figure B.2.2: Transceiver Block Diagram

[32] Nordic Semiconductors (2004). *Product Specification nRF9E5* (rev. 1.1). Retrieved October 31, 2006 from: http://www.nordicsemi.no/

Appendix C: Code

C.1 Acquisition Stage Code

The followings modifications were done to the SM_RX_VALIDATE state (Table C.1.1) in order to capture the entire Fence Line Communication Message and transmit it through the serial port.

Table C.1.1: Modified code on Commstask() module

```
switch (GetSMState())
{
case SM_RX_VALIDATE:
             message=FAIL_MESSAGE;
            GeneralSCISendChar(0x02);
            // 0x02 marks the start of text to send (0x03 is end)
            s=(unsigned char *)(&DataStart+7);
            // pointer positioned at the "length" octet of a FLC packet
            length=(int)*s;
            // variable length is equalized to address of the "length" octet
            s=(unsigned char *)&DataStart;
            // s goes back to the beginning of the FLC packet
            for (i=0; i<(length+10); i++)
            //  this cycle sweeps every the addresses of the
            // 10 assigned octets plus the entire length of the payload
                        {
                          GeneralSCISendChar(*s++);
                         // sends to the serial port each character from the
                         // read FLC packet increasing the pointer s until the cycle finishes
                        }   break;
}
```

Table C.1.2 shows the main program and how including only the relevant code for this application reduced it into a very simple program.

Table C.1.2: Modified Main program for the data acquisition stage

```
void main(void)
{
        // Initialisation routines for everything
        GeneralInit();
        WDInit();
        PowerOnReset();
        MemoryTest();
        SMInit();
        TimerInit();
        CommsInit();
        GeneralSCISetup();

        asm(" CLRC INTM");                  // enable global interrupts

        while(1)
                {
                        SMTask();           // upstate the state machine
                        TimerTask();         // update any timers that are running
                        CommsTask();        // do all needed communication tasks
                        WDTask();           // watchdog tasks enabled
                        if(AUTOFF==OFF)
                        activity=TRUE;      // stop auto off from occurring
                }
}
```

C.2 Transmission Stage Code

Commstask () function was modified as it is showed in Table C.2.1

Table C.2.1: Reprogrammed Commstask()

```
void CommsTask(void)
{
    unsigned char temp[40];
    char data;
    char result;
    char rxlength;

        data=SCIRXBUF;
        data=SCIRXBUF;
        data=0;
        rxlength=0;
        while(rxlength<10 || rxlength>38)
        {
                rxlength=GeneralSCIRecieveChar();
                while(rxlength==0xffff)
                {
                        rxlength=GeneralSCIRecieveChar();
                }
        }
        while(data < rxlength)
        {
                result=GeneralSCIRecieveChar();
                while(result==0xffff)
                {
                        result=GeneralSCIRecieveChar();
                }
                temp[data]=result;
                data++;
        }
        carrier=FSKDemodulation(dmCarrierSense); // see if line is free to use
        // Return value:  1 for no Carrier Sensed
        //                2 for Carrier was Sensed
        while(carrier!=1)
        {
                carrier=FSKDemodulation(dmCarrierSense); // see if line is free to use
        }
        LENdata_string=rxlength; // The number of bytes to transmit
        FSKModulation(temp); // Start to transmit
}
```

C.3 Radio Frequency Stage Code

The section of code framed on Table C.3.1, has the goal of defining the initial settings to configure the nRF95E IC.

Table C.3.1: Radio Frequency Initialization Routine

```
void RadioInit(void)
{
   unsigned char tmp;

   SPICLK = 0;              // Maximum SPI clock  1/2 of CPU Clock frequency
   SPI_CTRL = 0x02;         // SPI Connected internally to the nRF905 Radio Transceiver
   RACSN = 0;               // Transceiver Disabled
   SPI (RRC | 0x09);        // Read Configuration Register starting from byte 9
   tmp = SPI (0) | 0x04;    //  Write to Configuration register from byte 4, TX Payload
                            //  and assign to tmp
   RACSN = 1;                // Transceiver Enabled: pin P2.3
   RACSN = 0;                // Transceiver Disabled, ready for new instruction.
   SPI (WRC | 0x09);         // Write to Configuration Register starting from byte 9
   SPI (tmp);               // Write to Configuration Register from byte 4, TX Payload
   RACSN = 1;               // Transceiver Enabled: pin P2.3
   RACSN = 0;               // Transceiver Disabled, ready for new instruction.
   SPI (WRC | 0x03);       // Write to Configuration Register byte 3, RX Payload
   SPI (32);               // Write to Configuration Register byte 3, RX Payload
   SPI (32);               // Write to Configuration Register byte 3, RX Payload
   RACSN = 1;              // Transceiver Enabled: pin P2.3
   RACSN = 0;              // Transceiver Disabled, ready for new instruction.
   SPI (CC | 0x75);       //  80H+75H=F5H, meaning 11110101, CH_NO=868 Mhz  ,
                          // HFREQ_PLL=868 Mhz, PA_PWR(Power)= +10dBm.
   SPI (0x9f);
   RACSN = 1;              // Transceiver enabled: pin P2.3
}
```

SPI function (Table C.3.2) takes each input byte and moves it to the SPI data register.

Table C.3.2: SPI function

```
unsigned char SPI( unsigned char b )   // Exchanges one byte via the SPI interface
{
   EXIF &= ~0x20;                      // Clear SPI interrupt
   SPI_DATA = b;                      // Move byte to send to SPI data register
   while ((EXIF & 0x20) == 0x00)      // Wait until SPI has finished transmitting
      ;
   return SPI_DATA;
}
```

The following code (Table C.3.3) describes the UARTs Initialization Routine

Table C.3.3: UART Initialization Routine

```
void UartInit(void)
{
   TH1 = 230;             //9600@16MHz (when T1M=1 and SMOD=1)
   CKCON |= 0x10;         // T1M=1 (/4 timer clock)
   PCON = 0x80;          // SMOD=1 (double baud rate)
   SCON = 0x52;          // Serial mode1, enable receiver.
                         // Here SM2 or SCON.5 = 0 y REN=1
   TMOD &= ~0x30;
   TMOD |= 0x20;          // Timer1 8bit auto reload
   TR1 = 1;               // Start timer1
   P0_ALT |= 0x06;        // Alternate functions on pins P0.1-2
   P0_DIR = 0x02;         // P0.1 (RxD) in input
}
```

Table C.3.4 shows the ReceivePacket() code.

Table C.3.4: ReceivePacket() function

```
unsigned char ReceivePacket()
{
   unsigned char i;

   TRX_CE = 1;    // Enables reception. Writes in register P2 bit 5
   while(DR == 0) // DR: Data Ready notifies that the transmission is completed
       ;
   RACSN = 0;     // Transceiver disabled
   SPI(RRP);      // Read RX Payload
        for(i=0;i<32;i++)  // The 32 payload bytes of the packet are filled in buf
        {
           buf[i] = SPI(0);
        }
   RACSN = 1;     // Transceiver and ADC Chip select pin P2.3
   TRX_CE = 0;   // Reception Disabled: Writes to register P2 bit 5
   return 0;
}
```

The ReceiveChar() function makes use of the ReceivePacket() code (Table C.3.5).

Table C.3.5 ReceiveChar() function

```
void ReceiveChar(void)
{
    unsigned char b;
    unsigned char i;
    TXEN = 0;       // ShockBurst. Transmit-Receive selection : 0 = SET RECEIVER
    b = ReceivePacket();
    ES=1;           // Serial interrupt: TI or RI

          for(i=1;i<=buf[0];i++)
          {
                    while(!TI)
                    ;
             TI = 0;
          SBUF = buf[i];    // Sends to SBUF the received packet byte by byte
          }
}
```

Transmission code Tables C.3.6 , C.3.7 and main program on C.3.8

Table C.3.6: TransmitChar function.

```
void TransmitChar(unsigned char txlen, unsigned int offset)
{
   unsigned char i;
   TXEN = 1;           // ShockBurst. TX enabled
   RACSN = 0;          // Transceiver disabled, ready to write receive SPI instruction
   SPI(WTP);           // Instruction to Write to TX Payload register
   SPI(txlen);        //  Write length of the FLC message to TX Payload register
         for(i=0;i<txlen;i++)
         {
            SPI(buf[i+offset]);   // Write the rest of the FLC chars to TX Payload register
                                  // offset is a parameter of this function
         }
   RACSN = 1;         // Transceiver enabled
   TRX_CE = 1;       // Transmit packet stored in TX Payload register
   Delay(1);         // Delay until Shockburst process is finished
   TRX_CE = 0;      // Transmission process disabled
}
```

Table C.3.7: GetChar function

```
char GetChar(void)
{
         while(!RI)
                  ;
        RI = 0;
         return SBUF; // Listens and returns the character in SBUF if RI=1.
}
```

Table C.3.8: Main (Transmission program)

```
void main(void)
{
            unsigned char i;
            unsigned char len;
    while(1)
    {
                        Init();
                        len=GetChar();   // The first character sent is the FLC length.
                        if(len>15 && len < 49)  // FLC payload must be > 5 bytes and  < 39 bytes
                                                          // The biggest SMART PAC message (an Inquire)
                                                          // has 20 bytes of Payload.
                        {
                                    for(i=0;i<len;i++)
                                    {
                                         buf[i]=GetChar();  // The rest of the FLC message is sent
                                    }
                                    if(len<32)
                                    {
                                        TransmitChar(len,0);   // Typical Transmission
                                    }
                                    else    // if the FLC packet is bigger than 32 bytes (payload>21)
                                    {
                                         TransmitChar(31,0);        // TX the First 31 chars first
                                         TransmitChar(len-31,31); // TX from character 32
                                    }
                        }
     }
}
```

Appendix D: Message Types

D.1 Fence Line Communication Messages Types[33]

These are the different message types (Table D.1.1) defined for Fence Line Communication (FLC) packets.

Table D.1.1 : FLC Message Types

Number	Message Type	Description
0	ASNID	Assign ID: Used to Assign ID to Units
1	PING	To test the availability of a node on the network
2	ACK	Acknowledgment: Generated if the destination ID matches the local ID.
3	NACK	No Acknowledgment: Generated if the sent message has failed to get an acknowledgement reply with timeouts and retries completed
4	UNSUP	Unsupported: Generated if a received message has unsupported the message type
5	REQT	Request Type: Gets the unit type from a device. The reply should be of type DEVT
6	DEVT	Device Type: Is an ASCII string that specifies the unit type
7	X-PORT	XPORT: is an ASCII message type for sending XPORT protocol packets
8	DATA	Generic message for sending data
9	UDATA	Generic message for saying that last data message was not understood.

[33] Parker, S. (2003). *Protocol Specification FLC for GGL Fence Management System V0.3*. Hamilton: Gallagher R&D.