Willis, Simon L. (2007) ***Investigation into long-range wireless sensor networks.*** **PhD thesis, James Cook University.**

4 Node Design

This chapter discusses the design and testing of the long-range sensor node hardware. The chapter commences with a discussion of the design features and presents the radio transceiver hardware. The results of hardware simulations are first presented and these are supported with physical measurements shown later in the chapter. The design and testing of a suitable antenna is also discussed.

4.1 Design Features

A long-range wireless sensor node was developed called the JCUMote and is described by Willis and Kikkert in [61] (this paper is contained in Appendix C). The JCUMote was designed so that it has the inherent advantages of existing wireless sensor networks, such as being easy to deploy, robust, self-powered, self-configuring, energy efficient and versatile so that it can be used for a broad range of applications. The major design goal was achieving a long transmission range. This was realised by designing a novel radio transceiver that transmits 1 W of power at a frequency of 40.66 – 41 MHz. To operate in an ad-hoc network, each node must use a non-directional antenna, which has low gain compared to available directional antennas.

To reduce the software development time, it was decided to base the JCUMote design on the commonly used Mica2 Mote [18]. The Mica2 can be programmed with TinyOS and its schematic is publicly available on the TinyOS website [18]. A further discussion of TinyOS is included in section 5.2.

The JCUMote was designed to use the same expansion connector as the Mica2 to allow the use of the Mica2 programming board and sensor boards that are produced by Crossbow Technology Inc. [13]. Crossbow currently produces sensor boards to monitor temperature, humidity, light, sound, GPS position and acceleration.

The JCUMote was aimed to be used in rural Australian conditions and therefore must be installed in a robust enclosure. The Mica2 mote is powered by 2 AA batteries, which would give insufficient battery life on the JCUMote, therefore a larger battery is used. The design also allows for a solar panel to be installed, keeping the battery charged.

4.2 Node Structure

The structure of the JCUMote is shown in Figure 4-1. The node basically consists of a processor, flash memory, expansion connector and a radio transceiver.

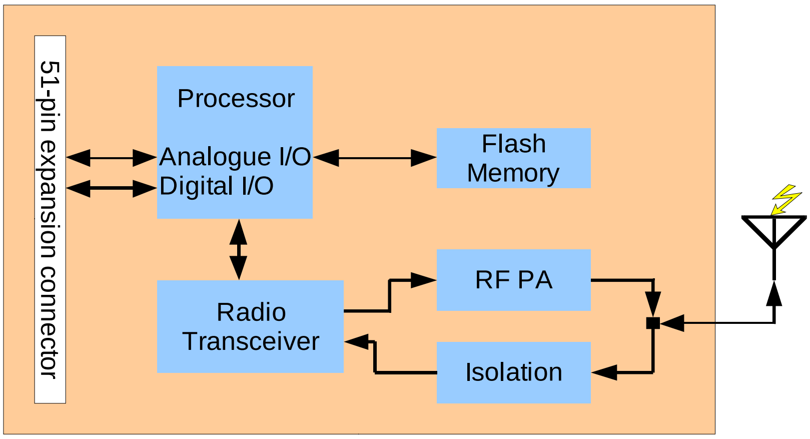


Figure 4-1: JCUMote Design

The JCUMote uses the ATMEL ATmega128L processor [19], which is an 8 bit Microcontroller with a 128 Kb self-programming flash program memory, 4 Kb SRAM, 4 Kb EEPROM and an eight-channel, 10 bit A/D-converter. The microcontroller unit (MCU) uses an ATMEL AT45DB041 4 Mb serial flash memory to store logged data and program images.

For radio communications a Melexis TH7122 [62] radio transceiver IC is used. This IC can operate at frequencies between 27 MHz and 1 GHz and emits 10 dBm (0.01 W) of power. The external PA boosts the output power to the required 1 W. When transmitting, an isolation network is activated to prevent the transmitted signal from damaging the receiver front-end of the TH7122. The receiver is rated to a maximum input power of +10 dBm and therefore requires the extra isolation.

4.3 Radio Transceiver

The Mica2 mote uses the Chipcon CC1000 radio transceiver chip which operates at frequencies of 315, 433, or 868/916 MHz with up to +10 dBm transmitted power. This transceiver is not suitable for operation at 40 MHz and therefore a suitable replacement was required. The Melexis TH7122 [62] was selected as it was the only transceiver IC that was found to be capable of 40 MHz operation. The TH7122 consists of a FSK/ASK modulator and demodulator, programmable interface and an output PA with +10 dBm maximum output. The block diagram is shown in Figure 4-2.

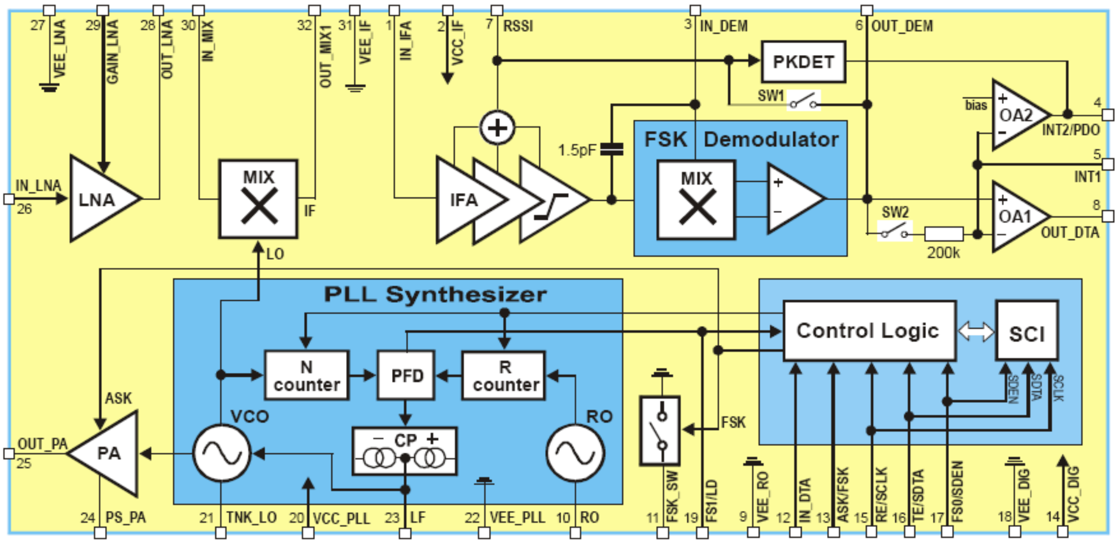


Figure 4-2: TH7122 Block Diagram [62]

4.3.1 Transceiver Design

The FSK modulation method was selected because the PA operates in class C mode, which requires an input signal with constant envelope. The radio transceiver was designed to allow for a maximum data rate of 19.2 kbps, which is the same as the Mica2. The frequency of modulation is twice this (38.4 kHz) as Manchester encoding is used. The system transmits a narrowband FM signal with a frequency deviation of 38.4 kHz. The modified Carson's rule (16) shows that the approximate bandwidth is 230 kHz, which shows that only one channel can be used if the system is operating at the maximum data rate. Alternatively, when the system has a data rate of 9600 bps, the channel bandwidth will be approximately 150 kHz, which is low enough to allow for two channels in the acceptable band.

$$BW_{\max} = 2(D+2)W \tag{16}$$

where D is the ratio of frequency separation to modulation frequency and W is modulation frequency.

Modulation is performed by injecting a signal into the voltage controlled oscillator (VCO) of the PLL.

For demodulation, the received signal is applied to a low-noise amplifier (LNA) and mixed down to the Intermediate Frequency (IF). An IF of 5.5 MHz was chosen instead of the conventional 10.7 MHz, as this allows the use of a low-cost narrowband (100 kHz) Murata IF filter designed for television sound reception. The TH7122 uses a standard FM quadrature detector for demodulation.

4.3.2 PHASE-LOCKED LOOP

The phase-locked loop is used to generate the carrier frequency for the transmitter and the local oscillator for the receiver. The block diagram for a phase-locked loop is shown in Figure 4-3.

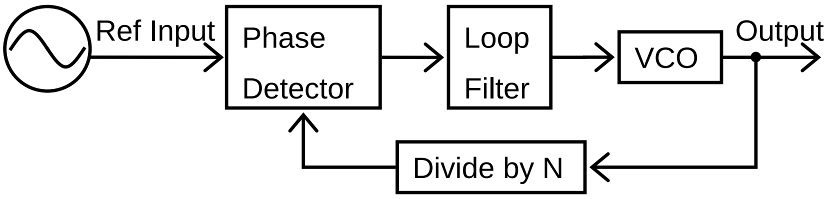


Figure 4-3: Phase-Locked Loop

The reference frequency is generated by an external crystal. This signal is compared to the feedback signal by the phase detector which produces pulses that relate to the phase difference between the two signals. The pulses are applied to the loop filter which is a low-pass filter that generates a DC voltage. This voltage is used to control the voltage controlled oscillator (VCO). The output signal is fed back to the phase detector via the divide by N circuit. Through the feedback system, the PLL generates an output signal with a frequency equal to N times that of the input.

The frequency of the output signal is determined by equation (17). Since frequency is the rate of change of phase, the output phase can be determined by equation (18).

$$F_{out} = K_V V_{VCO} \tag{17}$$

$$\phi_{out} = \frac{K_V}{s} V_{VCO} \tag{18}$$

where K_V is the gain of the VCO (Hz/V) and V_{VCO} is the VCO control voltage.
If the loop filter has a transfer function *G(s)* and the gain of the phase detector is defined as K_{Φ} then the transfer function for the PLL can be derived as shown in equation (19).

$$\frac{\phi_{out}}{\phi_{in}} = \frac{A}{1+AH} = \frac{K_\phi G(s)\frac{K_V}{s}}{1+\frac{1}{N}K_\phi G(s)\frac{K_V}{s}} = \frac{NK_\phi G(s)K_V}{sN + K_\phi G(s)K_V} = \frac{NKG(s)}{sN+KG(s)} \tag{19}$$

where $K = K_{\Phi}K_v$

The phase detector is usually made from logic circuits and produces a pulsed output. The loop filter must attenuate these pulses and present a smooth voltage to the VCO. Therefore, the loop filter must have a low-pass response with a bandwidth much less than the reference frequency. The loop filter must be chosen correctly as it has a strong impact on the function of the PLL.

The loop filter shown in Figure 4-4 has the transfer function defined in equation (20).

$$G(s) = \frac{s+a}{s(s+b)} \tag{20}$$

Equations (19) and (20) can be used to determine the transfer function for the whole PLL, defined by equation (21).

$$\frac{\phi_{out}}{\phi_{in}} = \frac{A}{1+AH} = \frac{K_\phi G(s)\frac{K_V}{s}}{1+\frac{1}{N}K_\phi G(s)\frac{K_V}{s}} = \frac{NK(s+a)}{sN(s+b)+(s+a)K} \tag{21}$$

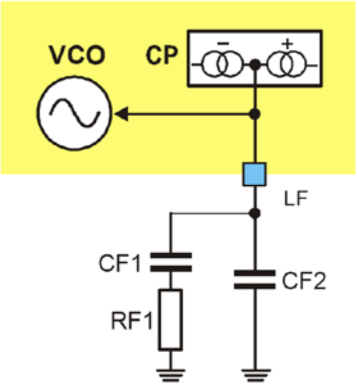


Figure 4-4: Loop Filter

4.3.3 Modulation

Modulation can be produced by a PLL using three methods: changing the N divider, changing the reference frequency or by combining the modulation signal with the VCO control voltage. The first method is commonly used for digital transmissions and requires the loop filter bandwidth to be greater than the data rate so that the PLL can respond to the frequency shift. The TH7122 does not allow the N divider to be changed at a fast rate and this method can therefore not be used.

The second modulation method is recommended by the manufacturer of the TH7122 [62]. The reference frequency is changed using crystal pulling which operates by switching the crystal load capacitor CX2 in response to the data (Figure 4-5). In order to achieve a frequency deviation of 38.4 kHz with a carrier frequency of 40 MHz, the operating frequency of the crystal must change by 1000 ppm. The datasheet for the 8 MHz crystal [63] used on the JCUMote specifies that the maximum frequency stability is ±100 ppm. Therefore, this method is not suitable.

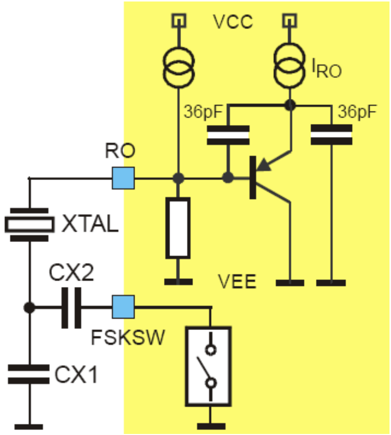


Figure 4-5: Crystal Pulling Circuit [62]

The third modulation method involves injecting a signal into the VCO. This method requires that the loop filter has a bandwidth that is much lower than the modulation frequency so that the PLL does not adjust to the modulating signal and cancel it out. A disadvantage of having a low loop filter bandwidth is that the PLL takes longer to adjust to a sudden change in frequency, such as switching from transmit to receive mode which involves reducing the frequency by an amount equal to the IF (5.5 MHz). A further restriction of this modulation method is that the modulating signal must not contain a DC component, as the PLL will adjust to the DC level and cancel out the modulation. This is not a concern when using Manchester encoding, as is the case on the JCUMote.

4.3.4 Modulation Circuit

The VCO (Figure 4-6) consists of a LC oscillator, which uses an internal varactor diode (VD) in parallel with an external inductor. The internal varactor diode acts like a voltage-controlled capacitor. A voltage increase on the TNK_LO pin causes a capacitance increase which lowers the VCO frequency. An external varactor diode has been added to allow for lower frequency operation. This diode, (VD1 and VD2 in Figure 4-7) is in parallel with the internal varactor diode and increases the overall possible change in capacitance of the resonant circuit, allowing for operation at lower frequencies.

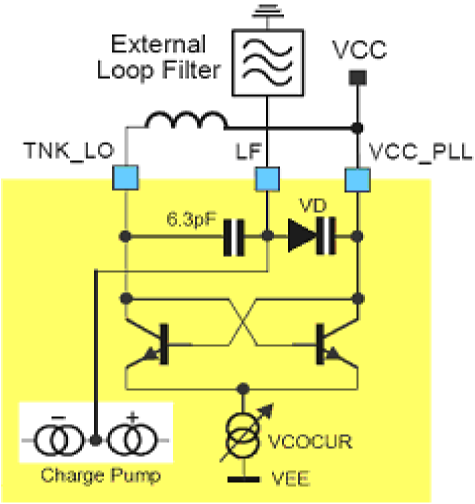


Figure 4-6: VCO Schematic [62]

Figure 4-7 shows the direct VCO modulation circuit, which was designed by the author and A/Prof C.J. Kikkert and is less complex than the circuit suggested by the manufacturer [62]. The circuit consists of a loop filter (components RF, CF1 and CF2), and a tank circuit (components L0, C0, VD1 and VD2) for the VCO. The modulating signal is applied to the circuit via CM1, RM1 and RM2 from the terminal 'MOD'.

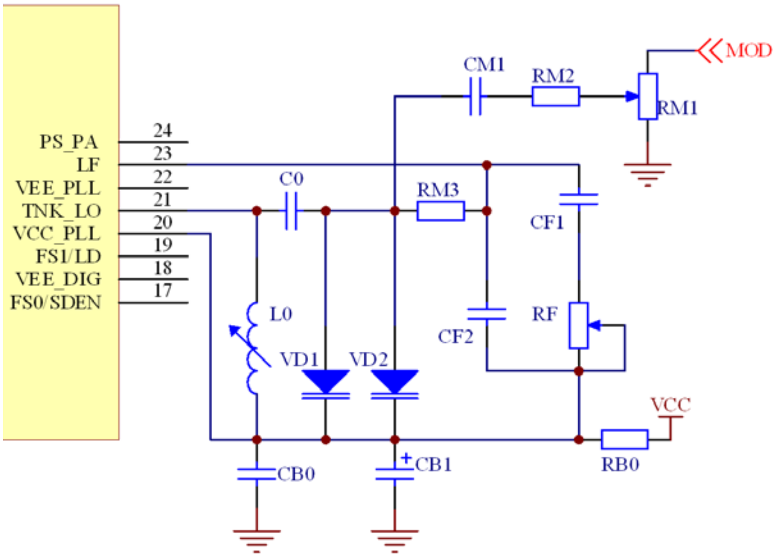


Figure 4-7: Direct VCO Modulation Circuit

The frequency of the VCO is controlled by the voltage across the varactor diodes (VD1 and VD2). The frequency deviation is set using RM1 and the modulation is combined with the loop filter control voltage via the RM2 and RM3 resistive adder.

The VCO is required to operate over a frequency range that is approximately 50 times the frequency deviation. Therefore, RM2 is approximately 50 times larger than RM3.

Since the PLL is also used as the local oscillator for the super heterodyne receiver, the frequency of VCO must be able to be reduced by 5.5 MHz, which is equal to the IF. Therefore, the loop filter components were chosen accordingly with an allowance given for a 5% temperature drift. To achieve the 5.5 MHz IF the carrier frequency must be changed by 14%. To accomplish a sufficient change in the capacitance of the resonant network, it was necessary to use two external varactor diodes in parallel (VD1 and VD2).

The loop filter was designed so that it has a bandwidth much lower than the lowest modulating frequency (9.6 kHz). This ensures that the PLL does not adjust to the modulation signal. The components' values were determined using equations provided by the manufacturer in [64]. It was stated in [64] that the bandwidth of the PLL can be approximated by (22).

$$\omega_T = \frac{K_{PD} K_{VCO} R_F}{N} \tag{22}$$

where K_{PD} is the gain of the phase detector, K_{VCO} is the gain of the VCO, R_F is the value of the loop filter resistor and N is the value of the N-divider in the PLL. This shows that the bandwidth is inversely proportional to N. Therefore, it was necessary to design the LF for the smallest value of N in Table 4-1, below.

4.3.5 DEMODULATION

The TH7122 has a super heterodyne receiver, which is illustrated in the upper part of Figure 4-2. The signal is received on the IN_LNA pin and amplified (LNA gain can be set with the control interface) before being applied to the mixer to generate the intermediate frequency (IF). The PLL generates the reference frequency for the mixer.

The IF signal is applied to an IF filter, which removes out-of-band signals. A low-cost narrowband 5.5 MHz IF filter (Murata SFSRA5M50BF00-B0) was chosen which is commonly used for television sound reception and has a 100 kHz bandwidth.

The IF signal is further boosted by the IF amplifier (IFA) which generates a received signal strength indicator (RSSI). The signal is applied to a quadrature detector, which phase shifts the signal by 90^o using a large reactance (1.5pF internal capacitor). A tuned external resonant network called the discriminator is used to contribute an additional phase shift to frequencies offset from the IF. The phase-shifted signal is mixed with the IF signal to give a DC voltage that is proportional to the phase difference. The Murata CDSRF5M50EK049-B0 ceramic discriminator was chosen as it has a 60 kHz bandwidth and will give a large phase change with the 38.4 kHz frequency deviation. The received data is output at pin OUT_DTA via a data slicer and op-amp.

4.3.6 Microcontroller – Transceiver Interface

The CC1000 on the Mica2 has a byte-level interface which allows a whole byte to be received from the MCU, Manchester encoded and then transmitted. Alternatively, the TH7122 has a bit-level interface which means that the data is transmitted bit-by-bit and must be Manchester encoded by the MCU.

It is desirable that the software radio interface is identical to that of the Mica2 so that the minimal changes are required to the TinyOS implementation of the MAC layer. To accomplish this task, the hardware interface was rewritten so that data was transferred with the data-link layer as bytes. To transmit, a byte is transformed into two Manchester encoded bytes and transmitted via the SPI. In order to achieve a low, selectable data rate the SPI is set to “slave mode” and the clock signal is generated by an output compare pin, as shown in Figure 4-8. The encoded data is output from the MISO (master-in, slave-out) pin.

Bytes are received on an input capture pin (IC1). To decode a byte, the received data is assembled on the MCU using a software algorithm, which was developed specifically for the JCUMote and is discussed in more detail in the software section (section 5.3.4).

The transceiver is programmed using a 3-wire synchronous serial system shown in Figure 4-8. The serial interface programs the control logic which allows the user to set the mode (receive, transmit, standby or idle), transmit and receive frequencies, gain of the LNA, output power and other parameters affecting the operation of the transceiver. The TH7122 requires inputs for serial clock (SCLK), serial data (SDTA) and serial data enable (SDEN). These signals are provided by the general purpose I/O pins: PD6, PD7 and PB2, respectively. The TH7122 also produces a lock detect signal that is set when the PLL is locked and is input to the MCU on pin PA6.

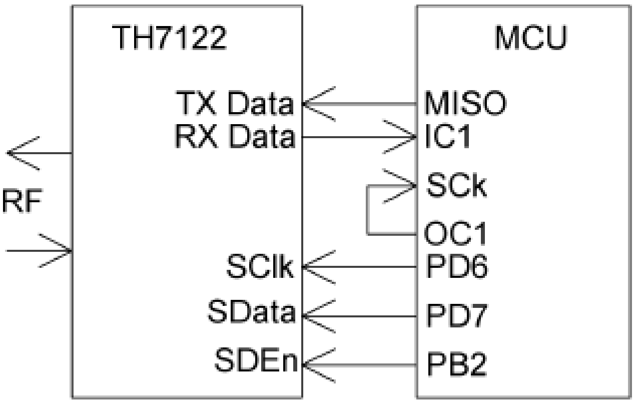


Figure 4-8: Microcontroller – Transceiver Interface

4.3.7 Channel Settings

Section 4.3.1 showed that 150 kHz channels can be used in the allowable 340 kHz of bandwidth. The transmit and receive frequencies of each channel are set by the N and R integer dividers in the PLL. The values for N and R are shown in Table 4-1

Table 4-1: Channel settings

Channel	Tx Freq.	NT	RT	Rx Freq.	NR	RR
1	40.75	815	160	35.25	705	160
2	40.9	818	160	35.4	708	160
3	40.85	817	160	35.35	707	160

If a data rate of 19.2 kbps is used then the Manchester encoder will output data at a rate of 38.4 kbps. The channel bandwidth can be estimated with the modified Carson’s rule (16) as 230 kHz. If this data rate is used, then channel 3 should be used to ensure that the signal does not appear outside of the allowable band. This frequency was used for all field tests (at 9600 bps).

4.4 External Power Amplifier

The external PA is required to boost the +10 dBm output signal from the transceiver to 1 Watt or +30 dBm. Since the amplifier is battery powered, it must be as efficient as possible, so Class C operation is required. In addition, the amplifier must be stable, cost effective and matched to the impedance of the transceiver IC and antenna. For 40 MHz operation, no suitable commercial amplifier IC is available, so the only cost effective approach was to design an amplifier using discrete components.

The amplifier circuit was simulated and tuned in Microwave Office [50] using non-linear Gummel-Poon models for the transistors. A large number of transistors were tested and in order to achieve the required gain, a two-stage amplifier circuit was found to be required.

The output transistor needed to be able to dissipate at least 2W of power, be able to supply a 500mA peak current, have a low ON resistance and have a gain of at least 10dB at 40MHz. The Zetex FZT649 [65] was selected as a suitable device. A drive stage is needed to provide sufficient drive for the FZT649 in class C operation under all temperature conditions. A MMBT2222A [66] general purpose transistor was selected for the drive stage and biased for class AB operation. Fig. 4 shows the circuit diagram of the RF power amplifier. The input impedance is tuned to give a full voltage swing on the open-collector output of the transceiver. The driver transistor uses an emitter feedback network for bias control. A matching network is used between the driver and the output transistor to provide the optimum gain and stability. A T-matching network is used at the output as this provides the required impedance transformation with the lowest losses in the output transistor. For stability, low resistance resistors are required at the base of each transistor.

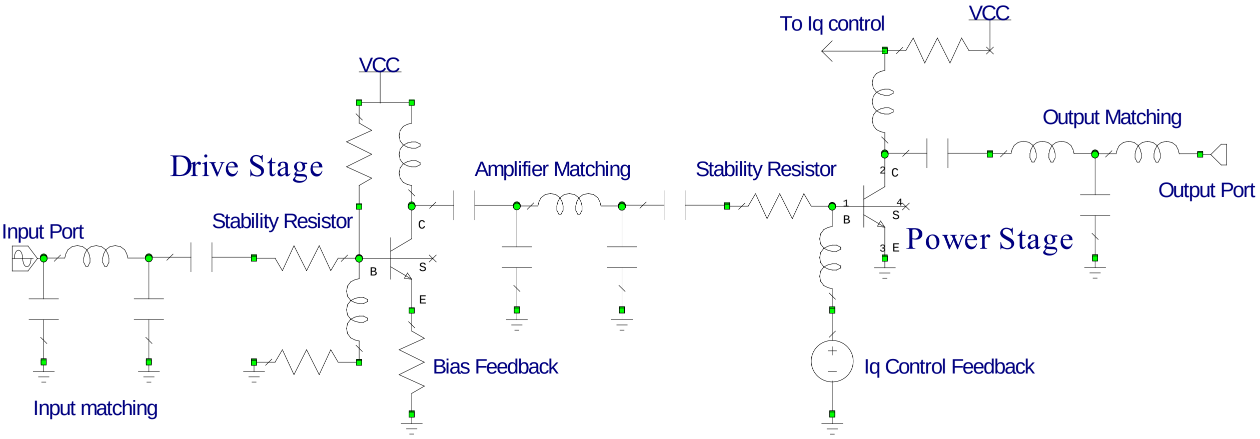


Figure 4-9: RF Power Amplifier Circuit [61]

4.4.1 COLLECTOR CURRENT CONTROL

The collector quiescent current is controlled using an op-amp feedback circuit. The circuit monitors the quiescent current via the current sensing resistor on the collector of the transistor. The input to the op-amp circuit is denoted 'To Iq Control' in Figure 4-9. This controls the quiescent current by adjusting the DC bias current denoted as 'Iq Control Feedback'.

Figure 4-10 shows the op-amp circuit, the 1 Ω resistor is the current sensing resistor placed in series with the collector. The gain and offset of the circuit can be varied by R_g and R_o, respectively. The circuit was configured so that the voltage of the op-amp is approximately half the supply (2.5 V) under normal operating conditions to allow for sufficient swing in the bias current.

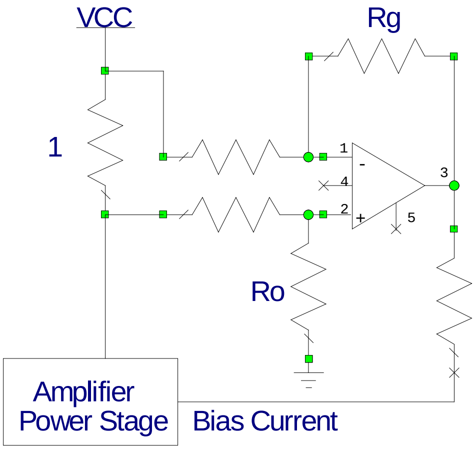


Figure 4-10: Collector current feedback circuit

4.4.2 GAIN AND DC TO RF EFFICIENCY

The amplifier circuit shown in Figure 4-9 was tuned and simulated in Microwave Office [50] to determine the gain and DC to RF efficiency shown in Figure 4-11. The available gain is the amplifier gain assuming ideal matching. Since there are losses in the matching networks, the actual gain (transducer gain) is slightly less. The gain shown in Figure 4-11 is larger than the 20 dB requirement. This allows for additional losses in the physical hardware. Figure 4-11 shows that the amplifier has approximately 48% efficiency. The initial design for the PA had much higher efficiency, but stability problems at low frequencies. The circuit was tuned to maximise stability, but some efficiency was sacrificed.

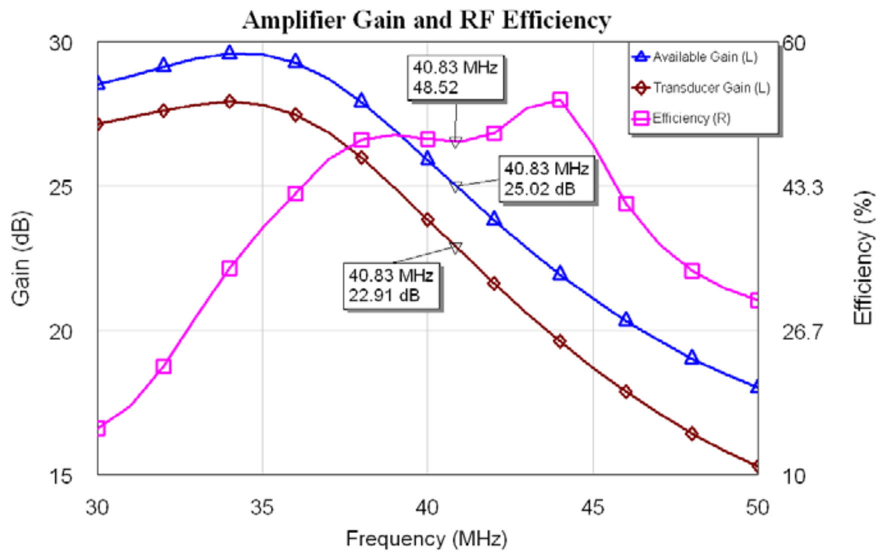


Figure 4-11: Available Gain and DC to RF Efficiency

4.4.3 AMPLIFIER POWER OUTPUT

Figure 4-12 shows the results from a non-linear simulation of the amplifier when a +10 dBm input is applied. The drive stage amplifies this to +19 dBm over the desired frequency range and causes a +31 dBm saturated output from the power stage, which is a sufficient margin above the +30 dBm requirement. The saturated gain of +21 dB is less than the +25 dB from Figure 4-11. The level of saturation is sufficient to ensure a constant power output with temperature and transistor parameter variations.

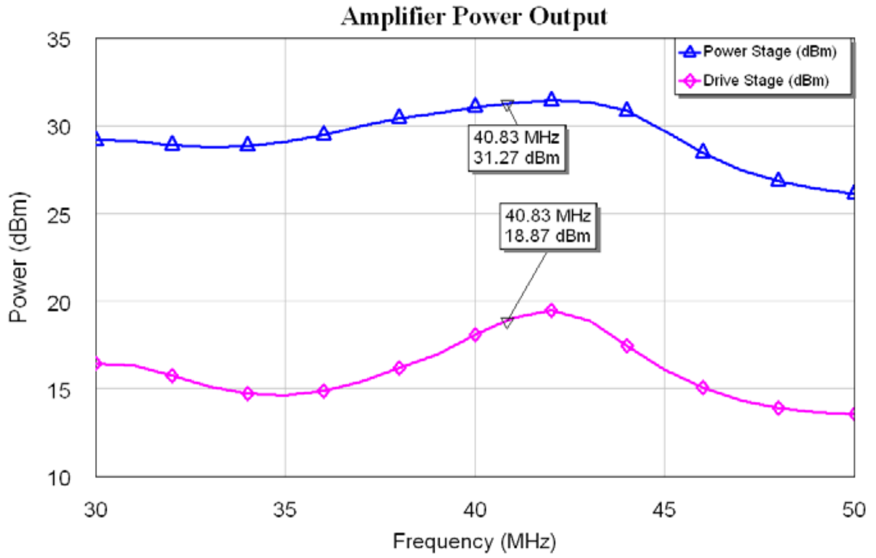


Figure 4-12: Amplifier Power Output

4.4.4 AMPLIFIER STABILITY

During the optimisation of the amplifier, the stability factor K, the supplemental stability factor B and the geometric stability factor μ were all determined over the frequency range of 5 to 70 MHz. For unconditional stability, K and μ must be greater than 1 and B must be greater than 0. From Figure 4-13, it can be seen that the amplifier is unconditionally stable over the desired frequency range. In order to achieve unconditional stability, small resistors were added to the bases of the transistors. These resistors are roughly 2 Ω each and improved the stability noticeably. These resistors have a negligible affect on the amplifier gain and efficiency.

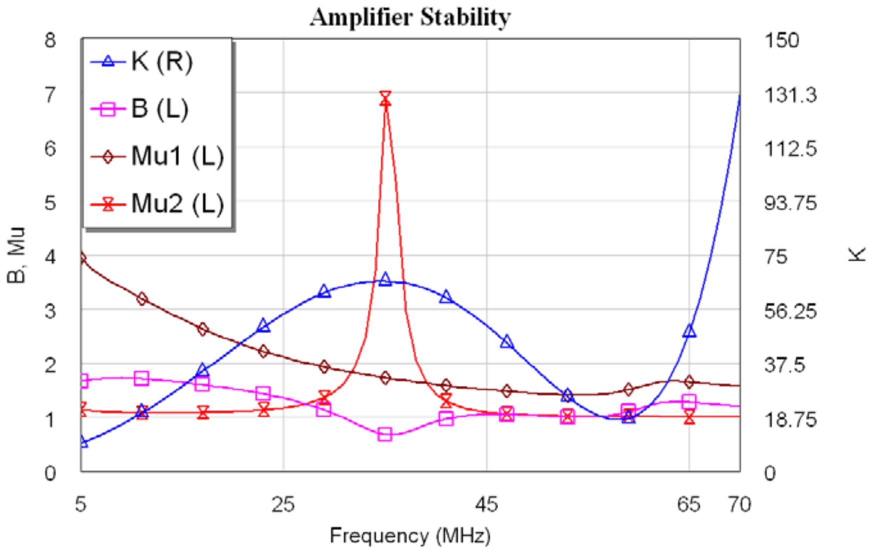


Figure 4-13: Amplifier Stability

4.4.5 TIME DOMAIN MEASUREMENTS

The time domain signals were simulated to determine the saturation of the amplifier and were also used as an aid when testing the PCB. Figure 4-14 shows the voltage waveforms when a 41 MHz +10 dBm signal (blue) is input to the amplifier. This signal has a voltage swing of 6 V_{pp} (Volts peak to peak), the maximum achievable output when using a 3 V supply. This shows that the amplifier is matched to the transceiver. The output of the driver stage (green) is 12 V_{pp}, showing the power stage is suitably matched to achieve full output swing.

The input impedance of the power stage transistor is low at 41 MHz (approximately 0.8 Ω) and this is reflected by the low 0.8 V_{pp} input at the base of the transistor (pink line, right axis). The transistor is turned on when the voltage on the base is above approximately 0.7 V. Figure 4-14 shows that this occurs less than half the time, corresponding to class C operation of the output transistor.

The voltage on the collector (brown) is 12 V_{pp}. After the impedance transformation, a 24 V_{pp} signal (red) is generated into the 50 Ω load. This corresponds to a RMS power output of +31.5 dBm (1.4 W) with a 50 Ω load, determined using (23).

$$P = \frac{V_a^{\ 2}}{2R_L} \tag{23}$$

where P is the power output in W, V_a is the amplitude (half of V_{pp}) and R_L is the load impedance.

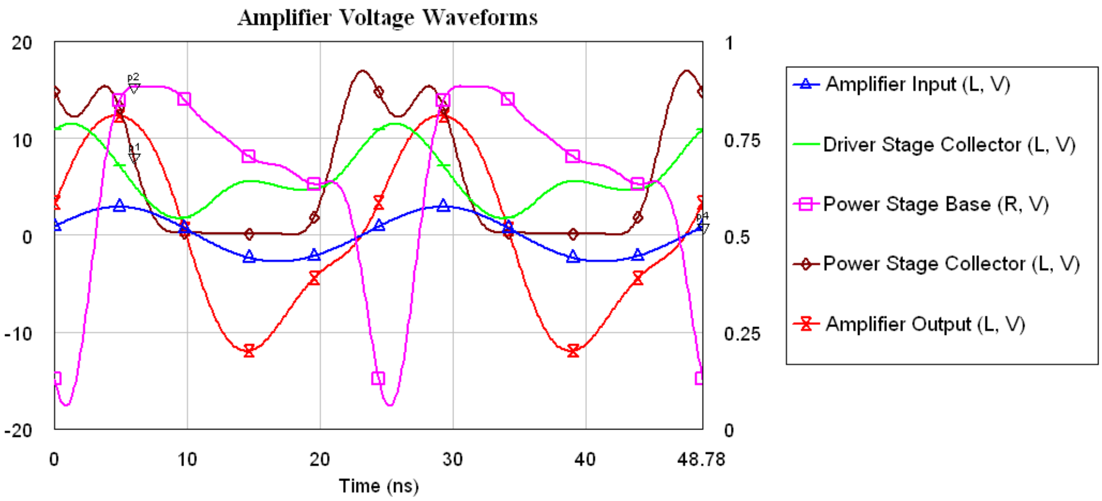


Figure 4-14: Amplifier Voltage Waveforms

4.5 Isolation Network

The TH7122 transceiver emits a small +10 dBm signal, which would usually cause no harm to the receiver front-end. However, the external PA boosts the signal to more than +30dBm which is potentially damaging to the receiver front-end. Hence, an external isolation network was added to protect the receiver when the node is transmitting. This network must attenuate the transmitted signal by at least 30 dB, but have minimal loss when receiving.

For efficiency, cost and reliability reasons, a solid state network rather than a relay controlled switch was used to provide the isolation. The solid state network must be able to handle the large voltages produced by the output amplifier. As a result, large signal non-linear simulation models were used to optimise the network.

Simulations in Microwave Office showed that when in transmit mode, the isolation network was able to attenuate a +30 dBm signal by 40 dB which is safe for the LNA. In receive mode, a small drop of 0.8dB is evident in the received signal. However, it should be noted that the effect of the power-stage transistor was not tested. When receiving, the transistor is in the off state, but the signals still reach the collector. It was difficult to simulate the transistor in the off state, so the effects of the collector are unknown.

The isolation network is shown in Figure 4-15. The voltage VPA is the power supply for the PA and is controlled by the transceiver. When in receive mode VPA is 0 V and the transistor is off. This forward biases D1 and D2 and reverse biases D3, hence allowing the signal to travel through the network. In transmit mode, VPA becomes 6V and the transistor turns on. This forward biases D3 and the voltage between D1 and D2 becomes 0.7 V, hence reverse biasing D1 and D2. Therefore, D1 looks like a high impedance and any current that passes D1 is shorted to ground via D3 and is blocked by D2.

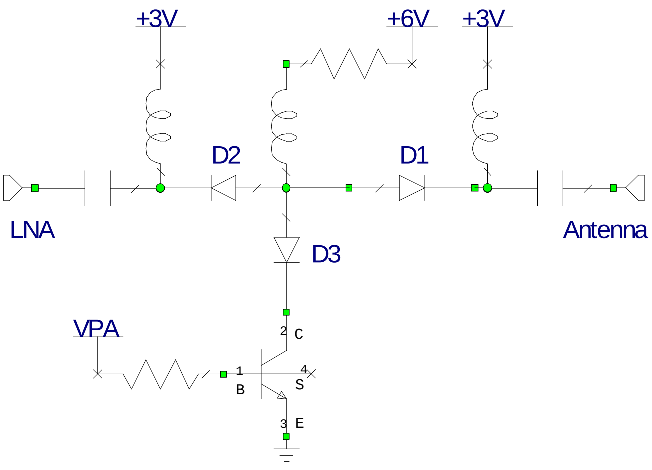


Figure 4-15: Isolation Network

4.6 SCHEMATIC

The schematic design is located in Appendix F and as discussed in section 4.2, is based around the Mica2 schematic that is available from the TinyOS web-site [18]. The JCUMote uses an ATMEL Atmega128L [19] microcontroller unit (MCU) which has a 10-bit eight channel A/D converter (ADC), a timer port with output compare and input capture, two USARTs, a SPI, an I^2C interface and several general-purpose I/O ports.

The MCU is connected to an ATMEL AT45DB041 4 MB dataflash memory IC, via a USART interface. This memory IC is used for datalogging and holding extra programs. A Texas Instruments DS2401P serial identification IC is connected to the MCU on a general purpose I/O pin and contains a unique ID number for each mote.

It was discussed in section 4.3.6 that the TH7122 uses the SPI for transmitted data and is connected to an input capture port for receiving data. On the Mica2, the input capture port (PD4/IC1) was originally used as the address latch pin for the 3-wire programming interface to the CC1000. The TH7122 uses a similar programming interface, but the address latch pin is now connected to PB2/MOSI, which was used on the Mica2 for receiving data and is hence not required on the JCUMote.

The Mica2 is powered by two AA batteries, which would be too small to power the PA on the JCUMote. Therefore, the JCUMote is powered by one 6 V battery which supplies the PA directly. A 3 V regulated source is used to power the MCU and sensor boards to retain compatibility with the Mica2 sensor boards. The regulated source is also used as the reference voltage for the analogue to digital converter (ADC).

Since a different supply voltage is used, it was necessary to alter the battery monitoring hardware. The Mica2 monitors battery voltage by using the reverse bias voltage of a zener diode as a reference. On the JCUMote, a voltage divider circuit is used which divides the battery voltage by 3 so that it is less than the 3 V ADC reference voltage. The circuit is shown Figure 4-16. VBAT is the battery voltage and

VSOL is the solar voltage. The circuit is activated by the BAT_MON signal which is produced by a general purpose I/O pin on the MCU.

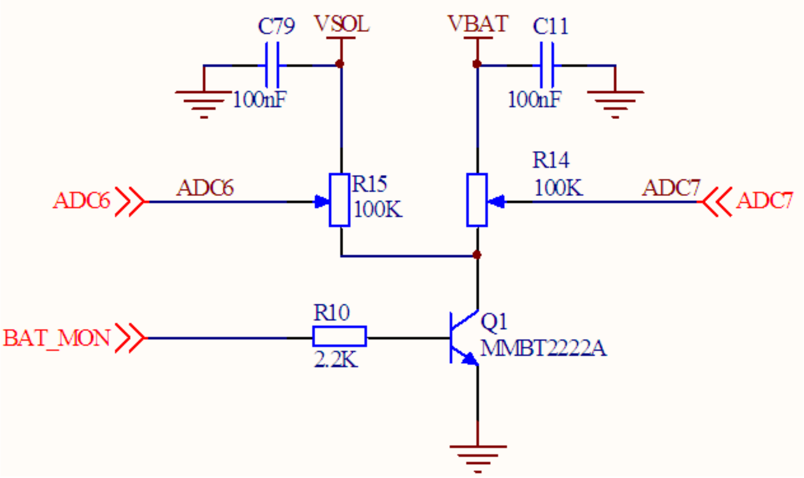


Figure 4-16: Battery Monitoring Circuit

4.7 Printed Circuit Board

The design of the printed circuit board (PCB) is described in Appendix G and the final product is shown in Figure 4-17. The PCB was designed, populated and tested by the author and printed by BEC Manufacturing in Brisbane, Australia. The PCB was designed to be small in size and is therefore double-sided. The tall components are located on the outside of the PCB to allow existing Crossbow sensor boards and programmer boards to be connected without fouling on the tall components.

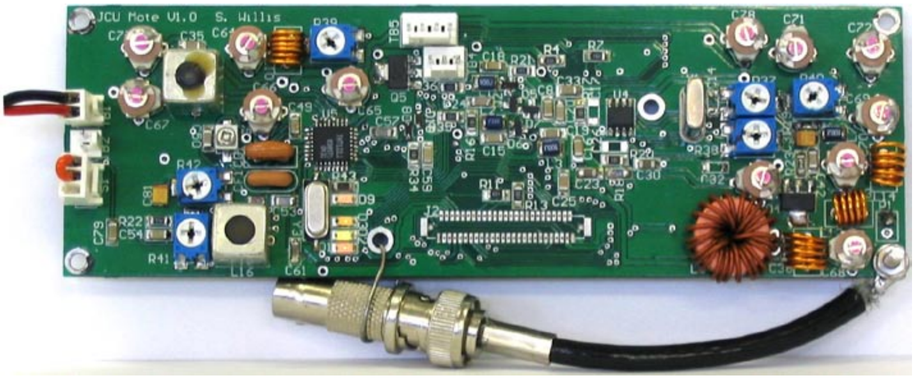


Figure 4-17: The top side of the PCB

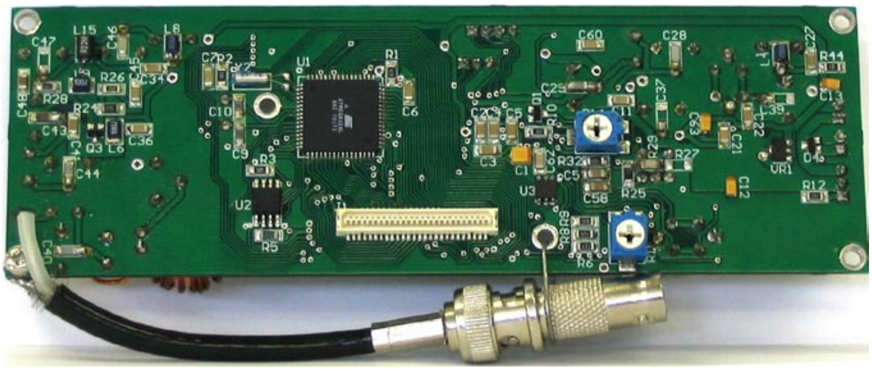


Figure 4-18: The bottom side of the PCB

To minimise RF noise, a ground plane was laid over the entire PCB. The PCB is mounted using the holes in each corner. The holes in the centre of the PCB are used for alignment of the sensor boards. Large planes were also placed around power components to dissipate heat.

4.8 Antenna Design

A number of antenna designs were tested for use with the LRWSN. This section starts with the specifications for a suitable antenna, followed by antenna theory. A number of antenna designs are presented and the results of testing are shown.

4.8.1 Specifications

For the LRWSN, an antenna must be non-directional to allow each node to communicate with surrounding nodes in any direction. The antenna must also be inexpensive and fairly small so that the node can be installed easily. Antennas that meet these two specifications typically have lower gain and therefore the high power output of the node is relied upon to achieve long-range communications.

It was envisaged that the sensor nodes would be installed close to, or on the ground, which is typically dry soil in an agricultural environment. This soil makes a poor ground plane, which must be considered when evaluating antenna designs.

4.8.2 Antenna Theory

An antenna can be modelled as an open-circuited transmission line as shown in Figure 4-19, where the open-circuit load (i.e. the end of the antenna) is on the left. If the antenna is a half-wavelength long, then a standing wave is produced and at resonance the current is in phase with the voltage. In this case, the antenna consumes all of the power and has low reactance.

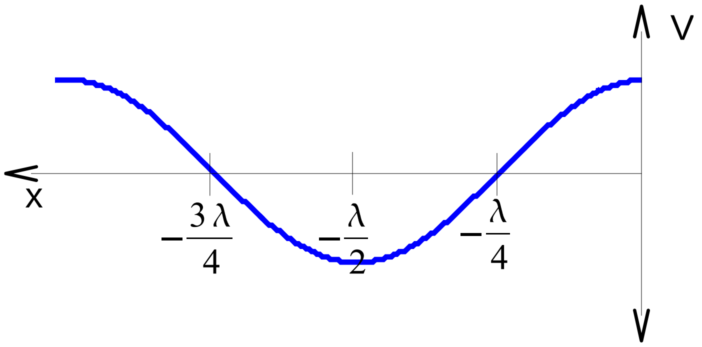


Figure 4-19: Open-circuit transmission line (load on left)

If the antenna is fed from the centre, then the waveform shown in Figure 4-20 is generated. This antenna is effectively made up of two quarter-wavelength antennas and is called a dipole. The impedance at the feed point is low and according to the ARRL Antenna Book, [48] is approximately 73 Ω in free-space. The impedance is dependent on the wavelength to rod diameter ratio, but is generally in the range of 50 to 75 Ω [48]. The ARRL Antenna Book [48] stated that the reactance is also dependent on the wavelength to rod diameter ratio and thicker conductors give the resonant circuit a lower Q. In practice, the length of a half-wavelength dipole is shorter than half the actual wavelength, because the signal travels at a lower velocity than the speed of light.

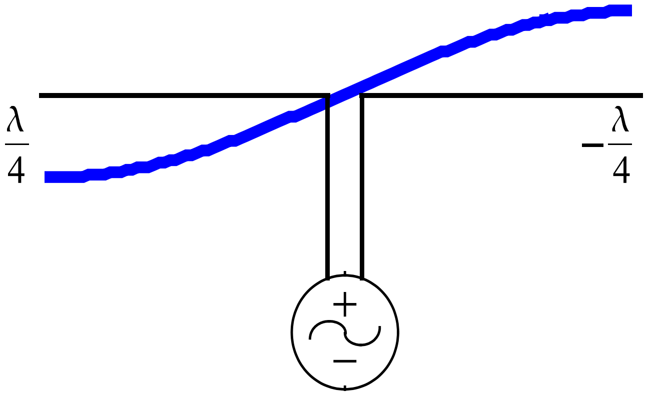


Figure 4-20: Half-wavelength Dipole

4.8.2.1 Grounding Systems

If a vertically polarised, quarter-wavelength antenna is positioned on a ground plane, then the ground acts as an electrical mirror to give a voltage waveform similar to that of Figure 4-20 and hence allows shorter antennas to be used. The length of the antenna can be further shortened by adding a loading coil at the base or a capacitive hat at the top of the antenna.

The LRWSN nodes will normally be positioned on the ground which is typically dry soil, a poor ground plane. The ARRL Antenna Book [48] stated that the optimum grounding system is 120 half-wavelength radial wires connected to the base of the antenna. However, if fewer wires are used then the length may be reduced. According to [48], it is sufficient to use 3-4 quarter-wavelength radials in most systems. If the radials are mounted on a 30° declination from the base of the antenna, then the impedance will be roughly 50 Ω.

4.8.3 ANTENNA DESIGNS

At the operating frequency of 40.66 – 41 MHz, the wavelength is 7.5 m. Hence a half-wavelength dipole would be 3.75 m long and is not practical for the nodes because it would be difficult to install. Quarter-wavelength antennas are easier to install due to their shorter length, but require a ground plane. A search was conducted to attempt to locate an antenna which is short and does not require a ground plane. The ARRL Antenna Book [48] shows many antenna designs that are commonly used for amateur radio, but none are presented that are sufficiently small and ground-independent. Figure 4-21 shows the J-pole, inverted-vee and folded dipole antennas that are commonly used ground-independent amateur radio antennas. The dimensions of these antennas are too large for use in the LRWSN.

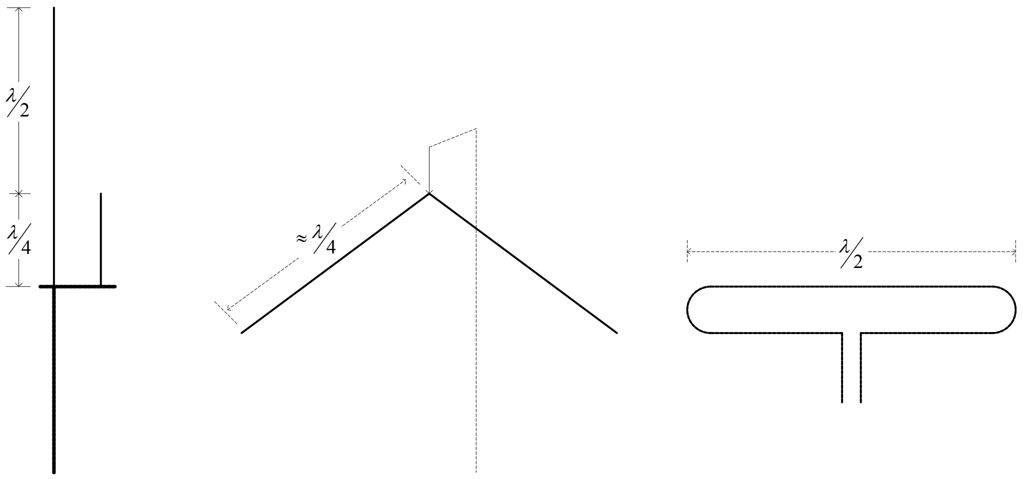


Figure 4-21: Commonly used amateur radio antennas [48]

A number of commercial antennas for the 40.66 – 41 MHz band were assessed ([67], [68], [69]). Centre-fed or folded dipole antennas are available at 40 MHz, however these are large and are designed to be permanently mounted with the centre feed point at least half a wavelength above ground. As an alternative, two suitable custom-made quarter-wavelength whip antennas (Mobile One [67]) were selected for testing: a 1.22 m helically wound whip and a 1.78 m braided antenna [67]. Both antennas were supplied with a standard threaded base connection to a separate base-plate. The antennas were less than $25 and were therefore, an inexpensive option. The disadvantage of using quarter-wavelength antennas is that they require a ground plane. The performance of these antennas was tested, along with a prototype design discussed below.

4.8.3.1 Prototype Antenna

The prototype design (Figure 4-23) consists of an Aluminium tube with the centre core of a coaxial cable running up the centre and attached at the top. The outer braid of the coaxial cable is attached at the base. This effectively forms a half-wavelength wire that is made up of the coaxial core and the aluminium tube. Since the base tube is grounded, there will be low voltage at the base and high voltage at the top of the antenna. This antenna also acts as a balun and it was expected to give a good impedance match.

For testing, the prototype antenna was installed on a base plate with a locking screw to allow the length of the antenna to be altered, as shown in Figure 4-22.

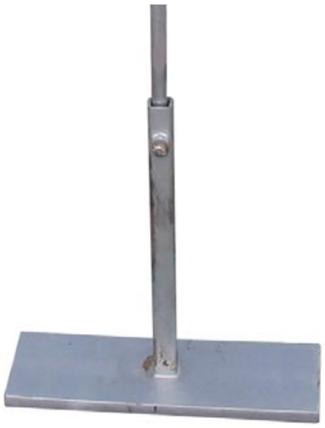


Figure 4-22: Base of prototype antenna

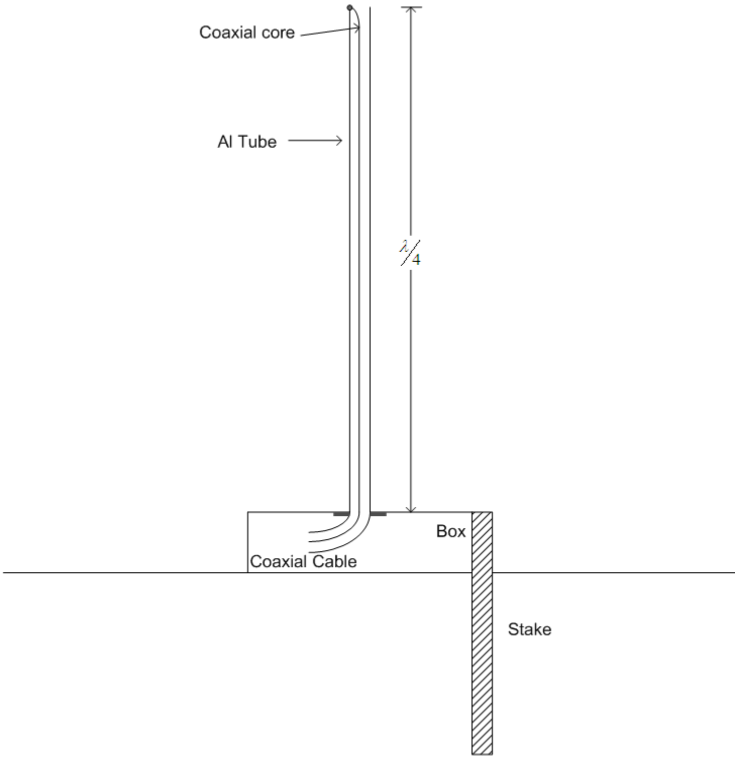


Figure 4-23: Prototype Antenna

4.8.4 RESULTS OF TESTING

The two Mobile One antennas and the prototype antenna were tested. Two tests were performed to determine the antenna impedance and gain. The first test involved measuring the impedance of each antenna at 40.83 MHz, which is the centre of the 40.66 – 41 MHz band. An impedance value of 50 Ω shows that the antenna will be matched well to the amplifier. If the impedance has a low reactive component, then this shows that most of the absorbed power is being transmitted by the antenna.

The second test was performed using an additional antenna to determine the gain of the antenna using the strength of the received and transmitted signals. Tests were first conducted on a roof top and later repeated on a grass area as this more closely matches the envisaged area in which the LRWSN would be installed.

4.8.4.1 ROOF TOP TESTS

The antennas were installed on an aluminium walkway on a building roof top as shown in Figure 4-24. For all tests, the base plate was clamped to the walkway. The impedance of the antennas was measured using a network analyser with a VSWR bridge.

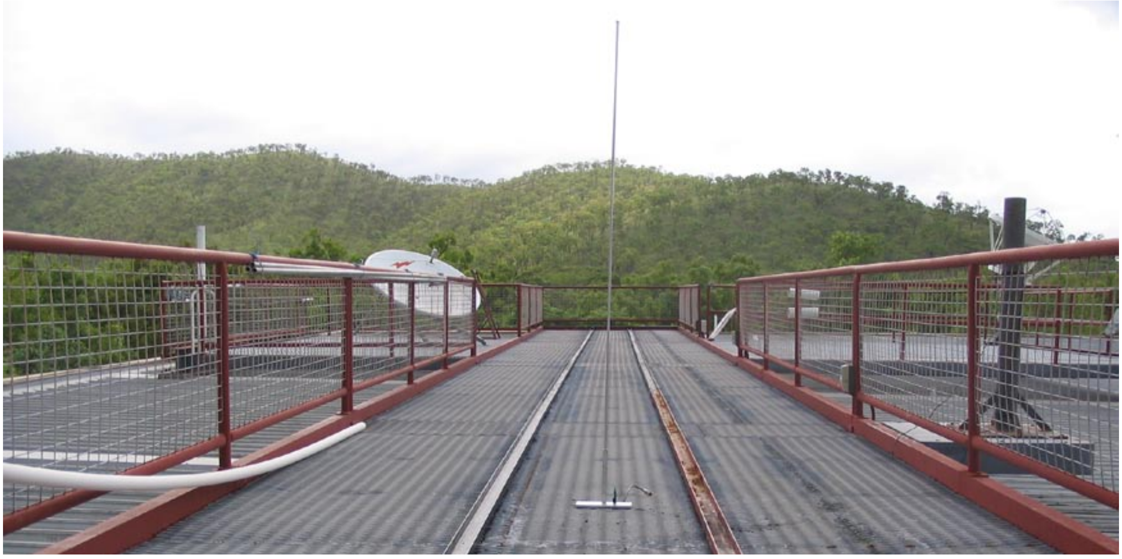


Figure 4-24: Prototype antenna installed on roof top

The impedance of the prototype antenna was measured as $1.7 - j48\ \Omega$ ($48\angle -88^\circ\ \ \Omega$) at 40.83 MHz. The reactance is much larger than the resistance which means that very little power is being radiated by the antenna. A second antenna was installed at a distance of 5 m from the prototype antenna and measurements on a spectrum analyser showed that no significant power was being radiated.

The prototype antenna behaved differently than expected, because it essentially acted as a short-circuited coaxial cable. With the rod grounded and the coaxial core running up the centre, the core coupled with the rod and behaved like a coaxial cable. This explains why the magnitude of the impedance (48 Ω) was close to the 50 Ω impedance of the coaxial feeder cable. The impedance phase angle of approximately -90° shows that the current is leading the voltage by 90°, as is expected for a quarter-wavelength short-circuited coaxial cable.

The prototype antenna was modified to improve the impedance match by tapping on to the Aluminium tube from the outside at a point of maximum resistance. This is shown in Figure 4-25. The impedance at this point was found to be 25 - j45 Ω at a distance of 250 mm from the base. The Smith Chart is shown in Figure 4-26.

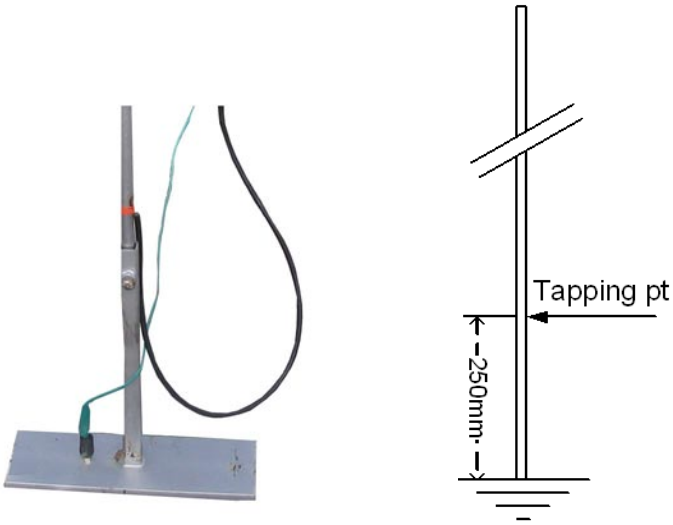
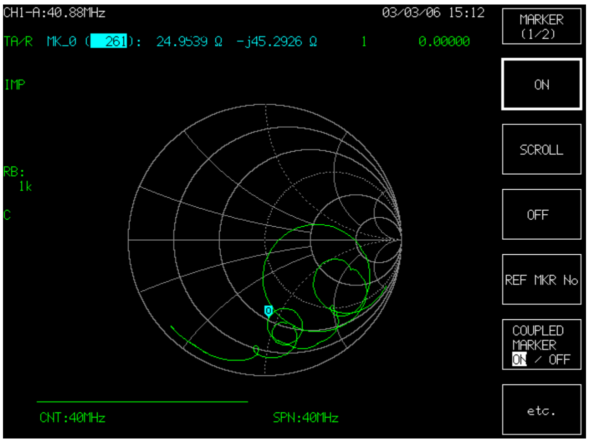


Figure 4-25: Antenna with tapping point



```
CH1-A:40.88MHz                                   03/03/06 15:12
TA/R  MK_0 ( 261): 24.9539 Ω  -j45.2926 Ω     1      0.00000
IMP
RB:
  1k
C
CNT:40MHz                        SPN:40MHz
```

Figure 4-26: Impedance of Al antenna at 250 mm from base

This impedance was matched to 50 Ω using a pi matching network that was calculated using the LCMatch program by Kikkert [70] and tuned using Microwave Office [50]. When installed on the roof top, the matching network was further tuned to give an impedance measurement of approximately 51 + j1 Ω, which is close to ideal.

The impedance of the commercial antennas was similarly tested using a network analyser. The antennas were installed on a base plate which was anchored to the walkway. The impedance was also measured (Table 4-2) on an aluminium roof, because it was envisaged that this would be a typical mounting scenario for a sensor node. The results showed that none of the commercial antennas had a good impedance match to 50 Ω.

TABLE 4-2: ANTENNA IMPEDANCE MEASUREMENTS

Antenna	Impedance (Ω)
Braided on aluminium roof	70 + j36
Braided on base plate	57 – j71
Helical on aluminium roof	97 – j58
Helical on base plate	32 – j68
Tapped prototype antenna with matching network	51 + j1

The receiving and transmitting gain of the antennas was tested using a second antenna and a spectrum analyser. The transmitting and receiving antennas were separated by a distance of 10 m (Figure 4-27). The transmitted signal was produced by an RF signal generator configured to output a +7 dBm signal at 40.84 MHz. An extra bandspanner antenna was used in the tests to act as a reference. The tests were conducted on three separate days to determine any variability in results.

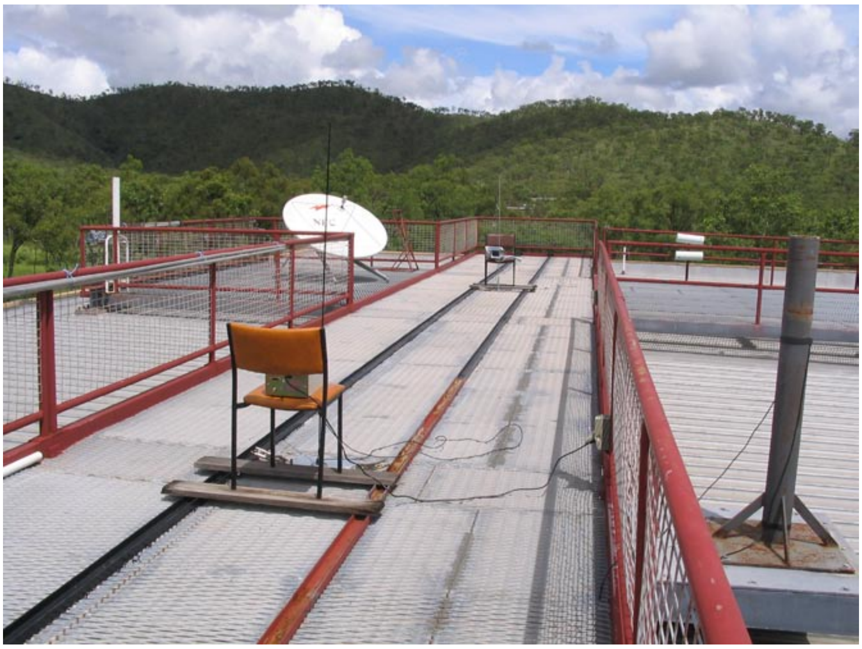


Figure 4-27: Antenna gain testing scenario

The results of all tests are shown in Table 4-3 ('M' indicates that a matching network was used). Table 4-3 shows that there is a large variation in the received signal strength between tests. In some cases, the signal strength had changed significantly between tests, such as 17 dB in the first row. The instruments, cables and anchoring to the ground plane were checked thoroughly, but no improvement in results was achieved.

It is inferred that the variability in the results was caused by multipath signals that were generated by reflections from the metallic objects on the roof. In addition to the variation in received signal strength, it was also noted that the antenna impedance changed dramatically on separate days. Due to the variation in the roof top tests, it was decided to disregard these results when determining the optimum antenna design. Further antenna tests were conducted on a grass surface as this more closely matches the envisaged sensor network environment and has less reflections from surrounding metallic objects.

TABLE 4-3: RECEIVED SIGNAL STRENGTH TESTS

		Signal Strength (dBm)		
Transmitter	**Receiver**	**Test 1**	**Test 2**	**Test 3**
Prototype (M)	Braided	-19	-36.1	
Prototype (M)	Helical	-20.8	-36.7	
Prototype (M)	Bandspanner	-27.6	-44.2	-29.9
Prototype	Braided	-18.9	-35.3	
Prototype	Helical	-20.1	-35.4	
Prototype	Bandspanner	-28.5		-30.8
Braided	Prototype (M)	-12.9		
Braided	Helical	-25.9	-22	
Braided	Bandspanner	-33.5	-29.4	
Braided (M)	Helical		-19.2	
Braided (M)	Bandspanner		-24.2	
Helical	Prototype (M)	-14.1		
Helical	Braided	-23.6	-20	
Helical	Bandspanner	-32.1		
Helical (M)	Braided		-18.1	
Helical (M)	Bandspanner		-25	
Bandspanner	Prototype (M)	-22.9		-37.7
Bandspanner	Braided	-33.6		-39.5
Bandspanner	Helical	-34.5		-40.2
Bandspanner	Helical (M)			-40.4
Bandspanner	Braided (M)			-40.7

4.8.4.2 GRASS ENVIRONMENT TESTING

The results of the roof top tests showed that the received signal strength measurements could not be relied upon as an indicator of antenna performance, because these measurements are largely affected by the surrounding environment. For the grass tests, the antenna impedance and reflection loss were used as an indicator of antenna performance.

It is envisaged that sensor nodes would typically be installed close to the ground, which is normally dry soil or vegetation, a poor ground. The ARRL antenna book [48] states that the grounding of the antenna has a large affect on the performance of the antenna. Therefore, it was important to investigate the grounding system used from the LRWSN nodes. Four different grounding systems were investigated: antenna positioned on the ground, antenna mounted at ground level on a 35 cm stake, antenna mounted on a 1.5 m steel picket and antenna mounted on a pole with 4 quarter-wavelength radials attached to the base, declined at an angle of 30^{o}. The latter testing platform is shown Figure 4-28. For all tests, the ground plate was physically attached to the picket.

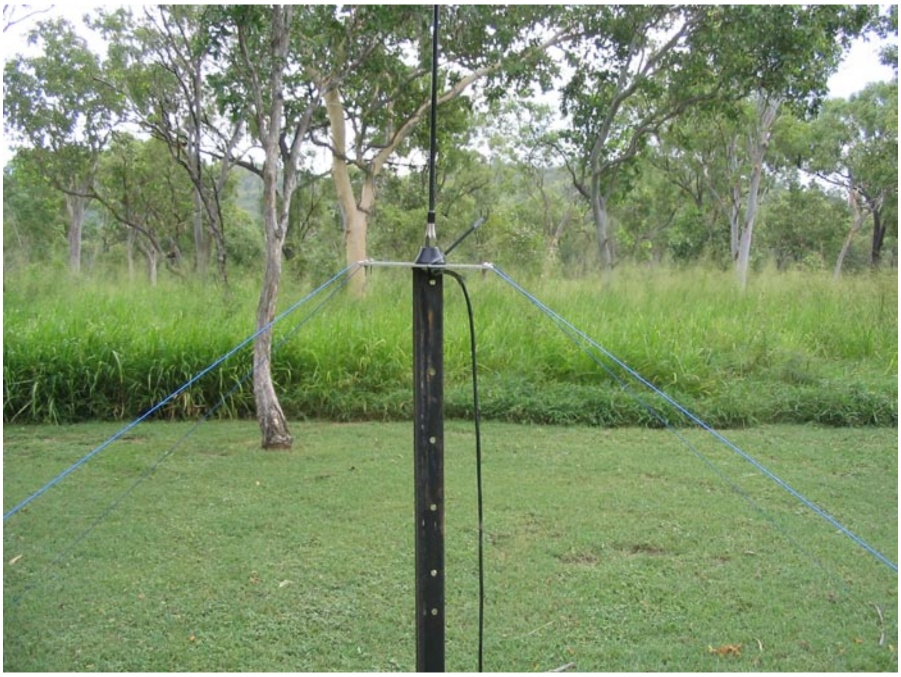


Figure 4-28: Picket with radials testing platform

The impedance and return loss were measured using a network analyser and VSWR bridge. It is desirable to maximise the return loss, as this means that most power is radiated and not reflected. Table 4-4 shows the results of the testing.

TABLE 4-4: RESULTS OF TESTING IN GRASS ENVIRONMENT

Grounding System	Antenna	Impedance at 40.84 MHz (Ω)	Return Loss at 40.84 MHz (dB)	Maximum Return Loss (dB)	Frequency of Maximum Return Loss (MHz)
Placed on grass	Braided	13.5 + j8.8	-4.6	-6	48
	Helical	15.6 + j9.8	-5.3	-	-
	Prototype (M)	13 + j16	-4	-14	45
	Prototype	18 – j54	-3.1	-	-
Ground level on 35 cm stake	Braided	34 – j6	-13.7	-13.9	39.2
	Helical	34 – j12.4	-12.5	-24.4	39.96
	Prototype (M)	14.1 – j26.8	-6	-21	43.92
	Prototype	13.4 – j50.3	-2.3	-	-
1.5 m picket	Braided	5.1 + j2.2	-1.78	-	-
	Helical	4.8 + j1.6	-1.7	-	-
	Prototype (M)	10.5 – j8.3	-3.6	-4.39	42.08
	Prototype	71 + j246	-0.8	-	-
Picket with 4 radials	Braided	48.1 – j9.6	-20	-27.35	40.32
	Helical	48.1 – j19.2	-14.2	-19	40.4
	Prototype (M)	8.2 – j16.5	-2.5	-3.65	42.56
	Prototype	45 + j209	-0.8	-	-

Table 4-4 shows that the picket with radials gives the optimum performance for return loss and impedance for both the braided and helical antennas. The Smith chart and return loss graph for the braided antenna are shown in Figure 4-29 and Figure 4-30, respectively. The impedance of this test platform is very close to the ideal value of 50 + j0 Ω and therefore supports the theory regarding radials for grounding as presented in the ARRL Antenna Book [48].

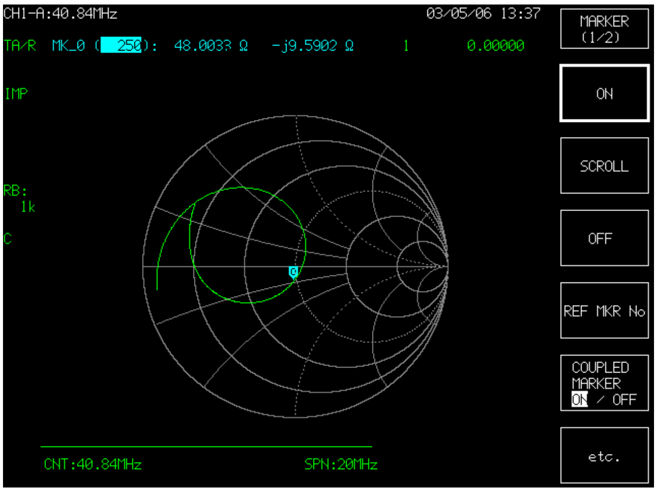


Figure 4-29: Impedance of braided antenna mounted on star picket with radials

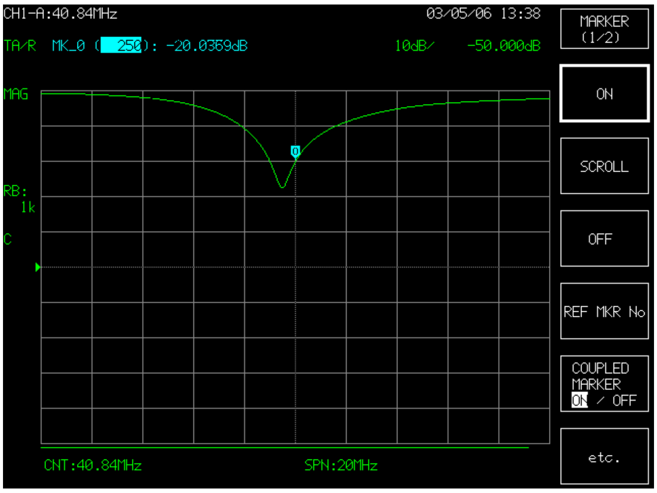


Figure 4-30: Return loss of braided antenna mounted on star picket with radials

The tests showed that the ground-height stake gave the second-best results, followed by the ground-height antenna with no stake and finally the picket-mounted antenna. The last result is interesting to note, as commercial antennas are commonly mounted in this fashion to achieve line of sight communications, however, the resulting impedance mismatch would reduce the gains associated with using a high antenna.

In addition to showing the optimum grounding system, the tests also showed that the braided antenna performed the best, followed by the helical antenna and the prototype antenna. The tests also showed that the matching network (indicated with an 'M') improved the return loss of the prototype antenna by at least 3 dB in most cases.

4.8.4.3 VERTICAL BAZOOKA ANTENNA

The vertical bazooka antenna is commonly used in amateur radio and is presented by Blake [71]. The antenna is made of coaxial cable, which has the braid rolled back over itself a quarter-wavelength. The antenna basically acts like a centre fed dipole with the coaxial inner element acting as the top radiator and the braid acting as the lower radiator. Additionally, the folded back braid interacts with the coaxial cable to form a balun. A diagram of the antenna is shown in Figure 4-31. In this case, the antenna is housed in plastic conduit and mounted to a post. A more practical mounting method for the LRSWN is to suspend the antenna from a tree.

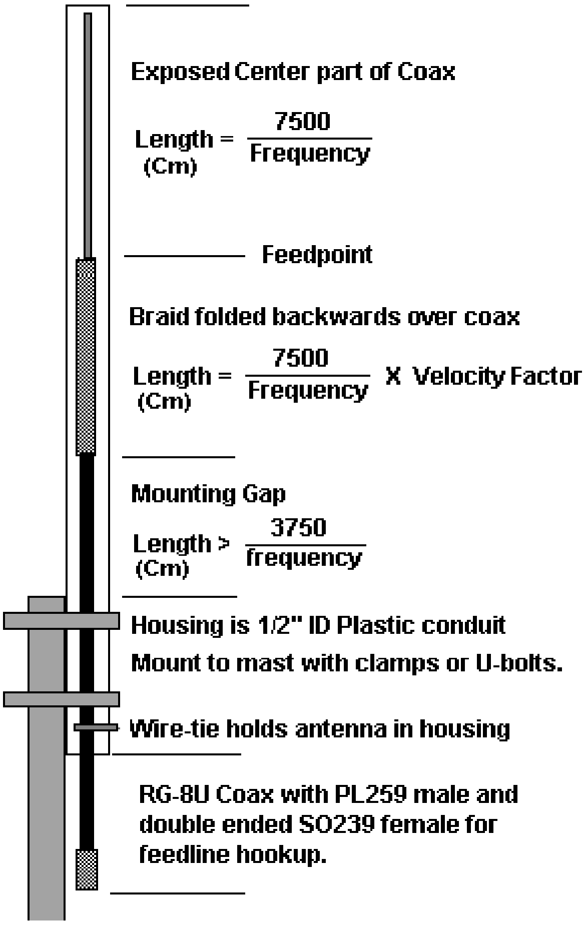

Exposed Center part of Coax
Length = 7500 / Frequency
(Cm)
Feedpoint
Braid folded backwards over coax
Length = 7500 / Frequency X Velocity Factor
(Cm)
Mounting Gap
Length > 3750 / frequency
(Cm)
Housing is 1/2" ID Plastic conduit
Mount to mast with clamps or U-bolts.
Wire-tie holds antenna in housing
RG-8U Coax with PL259 male and
double ended SO239 female for
feedline hookup.

Figure 4-31: Vertical Bazooka Antenna [71]

Blake [71] stated that the performance of the antenna is mainly affected by the length of the braid. To tune the antenna, Blake suggested adjusting the braid to optimise the return loss and then adjusting the length of the inner element. Blake also stated that the performance of the antenna can be improved by replacing the braid with a copper pipe.

Two different designs were tested using RG-58 and RG-8 coaxial cable. Each test was conducted above dry grass and the height was varied between less than quarter-wavelength to half-wavelength to find the best performance. It was found that neither the RG-58 nor RG-8 antenna performed as expected. The RG-58 coaxial had a maximum return loss of 10 dB at 44 MHz. However, the position of the braid seemed to have no affect on the frequency of the null.

A RG-8 coaxial cable was also tested. After tuning the braid and inner element, the best measured performance was a return loss of -5.76 dB at 40.87 MHz. The antenna was also tested with a copper pipe replacing the braid. This design had lower performance than expected with a -1.29 dB null in return loss at 34 MHz. Trimming the length of the copper pipe had no affect on the performance of the antenna. The Smith chart is shown in Figure 4-32 and illustrates that the antenna does not have a 50 Ω impedance at any frequency. Therefore this antenna is not suitable for the LRWSN.

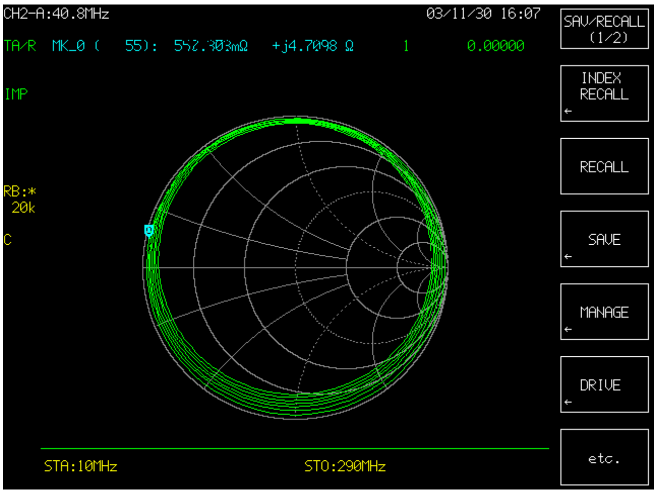


Figure 4-32: Smith Chart of Vertical Bazooka Antenna with Copper Pipe Balun

4.8.5 Conclusion of Antenna Testing

The rooftop tests showed that it is difficult to measure the transmit and receive gain of an antenna in an environment where there are signal reflections from surrounding objects. However, the results generally showed that the braided antenna performed the best. The prototype antenna was shown to give varying results depending on the surrounding environment and the matching network. It was decided that this antenna was too sensitive to changes in the environment to be used for the LRWSN.

The grass tests confirmed that the braided antenna gave the best performance. The investigation of the grounding systems revealed that the pole mounted antenna with radials gave optimum performance. However, this system is the most difficult to install. The ground-based tests showed that where possible, a stake should be used to improve the performance of the antenna system.

The Vertical Bazooka Antenna was tested as a practical alternative to the whip with radials antenna, but did not give the performance expected.

The braided antenna with four radials at a declination of 30^{o} was selected for use with the LRWSN as it gave the best performance and repeatable results.

4.9 Enclosure

A 140 × 120 × 80 mm enclosure was chosen to house the PCB and 6 V battery. A diecast aluminium enclosure was selected (Figure 4-33) as this is robust and acts as an RF shield. The enclosure is mounted with the lid on the bottom to prevent water entering. The antenna is mounted to the top of the enclosure and a stake is mounted on the side. The antenna radials are attached using the screw holes for the lid of the enclosure.

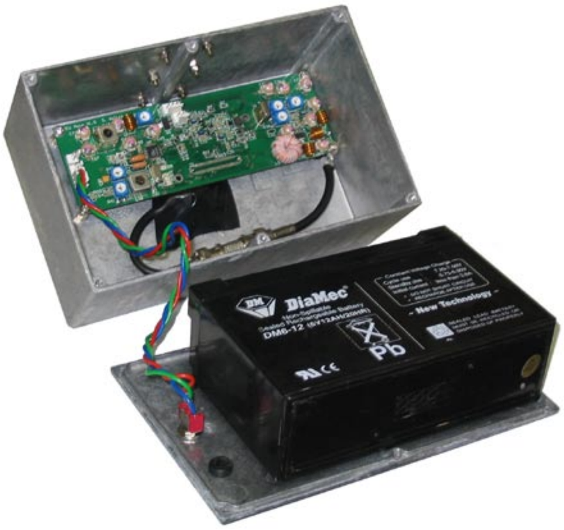


Figure 4-33: JCUMote Enclosure

4.10 Hardware Performance Measurements

The transmitter and receiver performance of each node was measured and the overall measurements are shown in Table 4-5. Node 1 was the first prototype PCB and had stability problems when outputting +30 dBm, so the transmitter power was reduced by adjusting the power-stage bias. This node also has lower receiver sensitivity. The cause of this is unknown, but is probably related to the quality of PCB, which was extensively modified from the initial design.

Table 4-5: Node Performance Measurements

Node	Transmitter Power (dBm)	Receiver Sensitivity (dBm)	Mean BER
1	+26	-70	2.93×10^{-3}
2	+30	-81	1.45×10^{-3}
3	+30	-79	1.66×10^{-3}
4	+30	-78	2.49×10^{-3}

4.10.1 Transmitter Measurements

The power output of the nodes was determined by measuring the output power spectrum. Figure 4-34 shows the power spectrum when a frequency deviation of 38.4 kHz is used with a 19.2 kbps data stream. It should be noted that a 20 dB attenuator was used whilst taking the measurement. This shows that the amplifier was outputting +30.5 dBm, which meets the requirement. The node draws 620 mA of current when transmitting. This corresponds to an efficiency of approximately 46%, which matches fairly closely to the 48% efficiency predicted by Microwave Office in section 4.4.2.

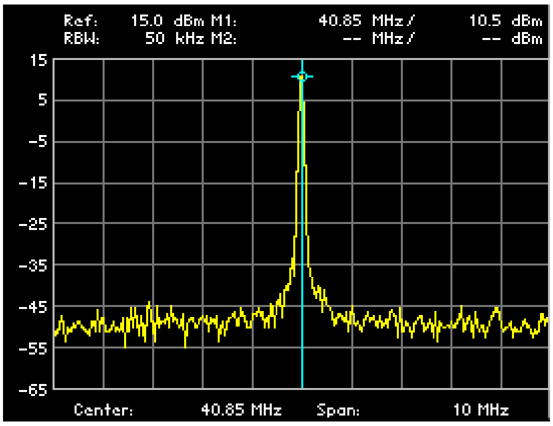


Figure 4-34: Measured Power Spectrum

To test the linearity of the modulation hardware, a triangle wave was input to the modulation hardware. A Rohde and Schwarz FSE spectrum analyser was configured to demodulate the signal. The resulting waveform (Figure 4-35), demonstrates that the modulator is fairly linear, because there is minimal rounding of the triangular wave.

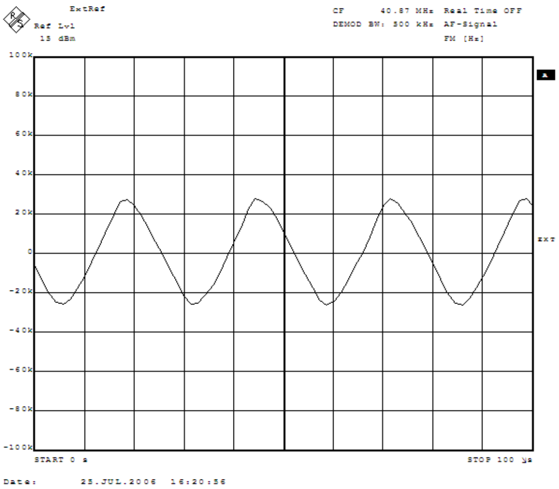


Figure 4-35: Reception of Modulated Triangle Wave

The receive-transmit switching time was measured by running a TinyOS program (`TxRxSwitch`), which switches between transmit and receive mode periodically. Figure 4-36 shows the lock-detect signal (yellow) and amplifier bias control voltage (blue). On the left of this figure, the node is in receive mode. The lock-detect signal drops whilst the node is switching from receive to transmit mode. The PLL switching time can be determined from the time taken for the lock detect signal to be set, which is 4 ms in this case. The blue line shows that the amplifier bias voltage takes approximately 6 ms to reach a steady-state. The radio transceiver control program (section 5.3.2) must wait for the bias voltage to settle before transmitting data.

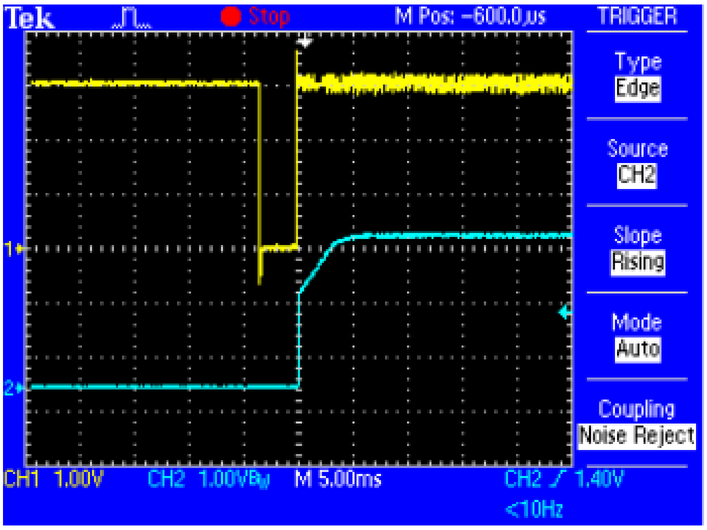


Figure 4-36: Receive-Transmit Switch Time

4.10.2 RECEIVER PERFORMANCE

The receiver sensitivity was determined using a Rohde and Schwarz AMIQ (I/Q Modulation Generator) and a Rohde and Schwarz SMIQ (Vector Signal Generator). The configuration of the test hardware is shown in Figure 4-37. The AMIQ generates a pseudo-random sequence which is modulated by the SMIQ and outputted to the node with known signal power. The demodulated signal from the node is then input to the AMIQ, which calculates the bit error rate and displays the value on a computer running Rohde and Schwarz WinIQSim software.

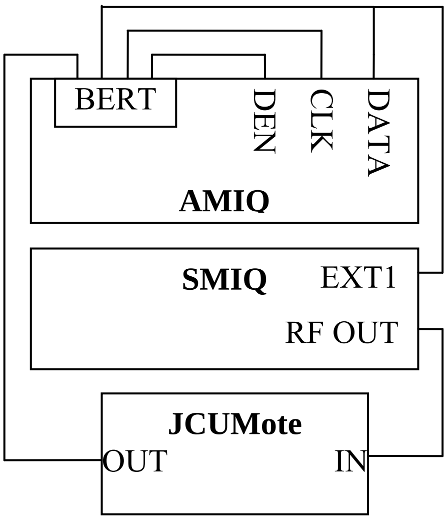


Figure 4-37: Receiver Sensitivity Test Equipment

The receiver sensitivity was determined by reducing the SMIQ output power until the bit error rate becomes greater than 3 × 10^{-3}. The resulting measurements are shown in

Table 4-5. For these measurements, a data rate of 20 kbps was used with a frequency deviation of 20 kHz.

It is stated in the TH7122 datasheet [62] that the receiver has a sensitivity of -105 dBm with a modulation frequency of 2 kHz and a frequency deviation of 100 kHz. Table 4-5 shows that the sensitivity of the JCUMote is less than this. It is inferred that the lower sensitivity is due to a number of reasons:

1. It is shown in section 4.10.2.2 that the ambient noise is significantly higher at 41 MHz. Therefore, a higher minimum signal requirement is expected.
2. The manufacturer's measurements were conducted with a lower modulation frequency (2 kHz) than that used with the JCUMote (20 kHz).
3. The manufacturer used a higher frequency deviation of 100 kHz, which improves the receiver sensitivity, but increases the signal bandwidth.
4. The manufacturer's measurements were conducted with a carrier frequency of 433.92 MHz and a 10.7 MHz IF. The manufacturer has not measured the performance of the receiver with a 5.5 MHz IF.
5. The manufacturer's test circuit was much simpler than that of the JCUMote. The manufacturer's test circuit did not require external varactor diodes because it was designed for a higher carrier frequency. The use of external varactor diodes on the JCUMote is a possible source of noise into the PLL.
6. The manufacturer's test circuit did not require the external PA and isolation network. These are a possible source signal attenuation in receive mode.
7. The JCUMote uses three matching networks between the antenna and mixer (PA output matching network, receiver input matching network, LNA to mixer matching network). Each network is a possible source of attenuation. The manufacturer's test circuit only required the PA output matching network.
8. The JCUMote also has a long signal path between the antenna and the receiver on the PCB. This can cause attenuation of the received signal and can also be affected by noise.
9. The manufacturer's test circuit did not include any additional digital circuitry, whereas the JCUMote has the Microprocessor and relating circuitry on the same PCB. These are a possible source of noise.

It is expected that the main reason for lower sensitivity is due to signal losses caused by the PA and isolation network as well as the matching networks.

4.10.2.1 Noise Analysis

A noise analysis was conducted to further investigate the receiver performance. The aim of this analysis was to determine the TH7122 receiver noise figure from the manufacturer's test results and compare this with the noise figure of the JCUMote.

The Melexis specifications [62] state that the TH7122 has a typical sensitivity of -105 dBm for a BER $\leq 3 \times 10^{-3}$ and with an IF bandwidth of 150 kHz. Measurements were conducted at 23°C with a carrier frequency of 433.92 MHz. An IF of 10.7 MHz was used.

To achieve a BER of 3×10^{-3} a SNR of 8.78 dB is required. This was determined using equation (24), which was derived from equation (13).

$$SNR = 2\left(\operatorname{erfc}^{-1}(2BER)\right)^2 \qquad (24)$$

where *SNR* is the signal-to-noise ration, erfc^{-1} is the inverse of the complementary error function and *BER* is the bit error rate.

With a bandwidth of 150 kHz and a temperature of 23°C, the thermal noise is -122.1 dBm as determined using equation (25):

$$N_T = kTB = -122.1 \text{ dBm} \qquad (25)$$

where N_T is the thermal noise, *k* is Boltzmann's constant, *T* is the temperature (in Kelvin) and *B* is the bandwidth (in Hz).

The noise figure is determined using equation (26) which subtracts the thermal noise and the receiver noise figure from the sensitivity level (P_R).

$$NF_{rx} = P_R - SNR - N_T \qquad (26)$$
$$NF_{rx} = -105 - 8.78 + 122.1 = 8.32 \text{ dB}$$

where P_R is the received power (sensitivity), SNR is the signal-to-noise ratio and N_T is the thermal noise.

Therefore, the noise figure of the TH7122 according to the Melexis specifications is 8.32 dB.

The JCUMote has a sensitivity of -81 dBm to achieve an average BER of 1.45×10^{-3}. This requires a SNR of 9.48 dB (24). The JCUMote uses a 100 kHz IF filter which gives a thermal noise level of -123.8 dBm. The noise figure of the JCUMote can be calculated by equation (27).

$$NF_i = P_R - SNR - N_T \tag{27}$$
$$NF_i = -81 - 9.48 + 123.8$$
$$NF_i = 33.3 \text{ dB}$$

where N_T is the thermal noise, NF_i is the noise figure of the TH7122 receiver circuitry, SNR is the signal to noise ratio and P_R is the receiver power.

Equation (27) shows that the noise figure of the JCUMote is 25 dB higher than the standard TH7122 receiver circuit. Earlier in this section, it was identified that this is likely to be caused by signal loss in the matching networks, PA and isolation network, as well as the use of a lower IF.

4.10.2.2 Measuring Environmental Noise

The receiver performance is limited by the level of environmental noise. To measure the environmental noise level in a suburban area, a quarter-wavelength whip antenna was connected to a Rohde & Schwarz FSL Spectrum Analyser. When measuring low signal levels such as background noise, it is important to account for the noise figure of the measuring equipment. This section discusses the method of determining this and presents the measured environment noise.

The noise figure of the spectrum analyser was determined by measuring the noise level when terminated with a 50 Ω resistor. Figure 4-38 shows that the noise level is -144.62 dBm/Hz.

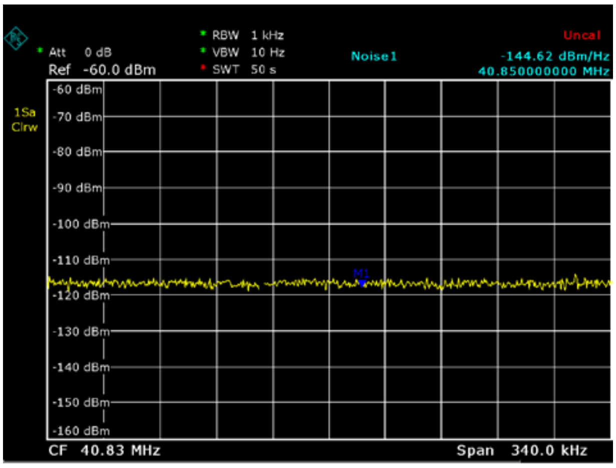

Att 0 dB
Ref -60.0 dBm
RBW 1 kHz
VBW 10 Hz
SWT 50 s
Noise1
Uncal
-144.62 dBm/Hz
40.850000000 MHz
CF 40.83 MHz
Span 340.0 kHz

Figure 4-38: Noise Level when terminated with a 50 Ω load

The thermal noise of a 50 Ω resistor at room temperature is -174 dBm/Hz. Therefore, the spectrum analyser has a 30 dB noise figure.

To reduce the noise figure, two 20 dB low noise amplifiers (LNA) were added to the input of the spectrum analyser. These LNAs use a MiniCircuits MMIC GALI-S66+ amplifier which is quoted to have a noise figure of 2.05 dB at 50 MHz [72]. The gain of the amplifiers was measured as 20.5 dB and 20.4 dB.
The overall noise figure can be determined using Friis's equation (28) as shown in [55]

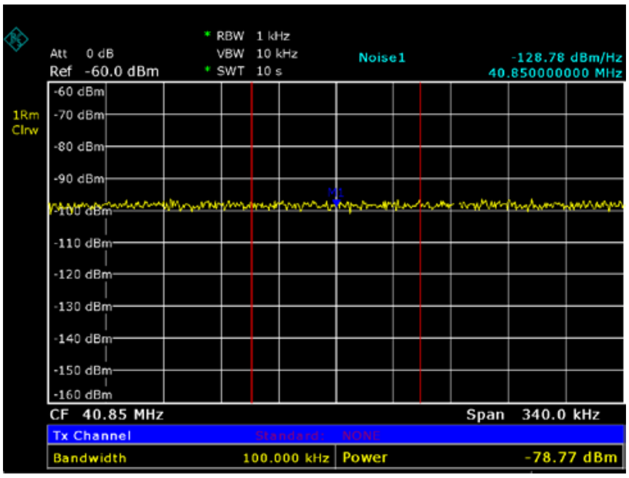
$$F_T = F_1 + \frac{(F_2 - 1)}{G_2} + \frac{(F_3 - 1)}{G_1 G_2} \tag{28}$$

where F_T is the total noise figure, F_n and G_n are the noise figure and gain of the nth amplifier, respectively.

Using the values above, the total noise figure was found to be 2.17 dB. When terminated with a 50 Ω resistor, a noise level of -130.9 dBm should be measured, as shown in equation (29).

$$P_N = N_T + G_1 + G_2 + F_T \tag{29}$$
$$P_N = -174 + 20.5 + 20.4 + 2.17$$
$$P_N = -130.9 \text{ dBm}$$

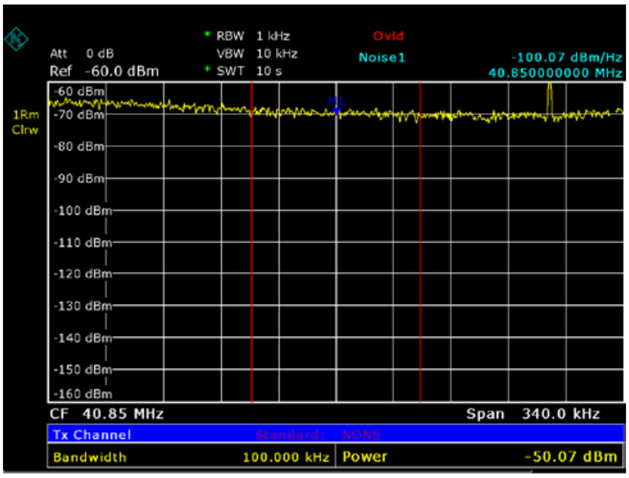
where P_N is the measured noise and N_T is the thermal noise of the resistor (-174 dBm/Hz). The measured value was -128.8 dBm/Hz as shown in Figure 4-39, which also shows that the total power in the 100 kHz channel is -78.7 dBm. This measured noise value is 2.13 dB higher than expected. This is likely to be caused by extra noise in the system that was not measured.



Att 0 dB
Ref -60.0 dBm
RBW 1 kHz
VBW 10 kHz
SWT 10 s
Noise1 -128.78 dBm/Hz
40.850000000 MHz
CF 40.85 MHz
Span 340.0 kHz
Tx Channel Standard: NONE
Bandwidth 100.000 kHz
Power -78.77 dBm

Figure 4-39: Measured Noise and Channel Power with a 50 Ω Termination

When the antenna was connected to the amplifiers, the noise level and channel power was measured and is shown in Figure 4-40. This figure illustrates that the signal level is 28 dB higher than the resistor noise level.



Att 0 dB
Ref -60.0 dBm
RBW 1 kHz
VBW 10 kHz
SWT 10 s
Ovld
Noise1 -100.07 dBm/Hz
40.850000000 MHz
CF 40.85 MHz
Span 340.0 kHz
Tx Channel Standard: NONE
Bandwidth 100.000 kHz
Power -50.07 dBm

Figure 4-40: Noise Level and Channel Power with Antenna

The actual atmospheric noise level was calculated using equation (30).

$$P_{env} = P_N - F_T - G_1 - G_2 - NF_e \tag{30}$$

$$P_{env} = -50.07 - 2.17 - 20.5 - 20.4 - 2.13$$

$$P_{env} = -95.3 \text{ dBm}$$

where P_{env} is the environmental noise power, P_N is the measured noise power and NF_e is the noise figure error factor discussed above.

Substituting the -50.1 dBm channel noise power value into (30) shows that the environmental noise is -95.3 dBm. If a carrier-to-noise ratio of 8.78 dB is required to obtain a BER of 3×10^{-3}, then the magnitude of the carrier must be greater than -86.5 dBm. Table 4-5 showed that the JCUMote has a receiver sensitivity of -81 dBm, hence a further improvement of 5 dB would improve the range of the node, but further improvements in sensitivity would have little affect due to the high levels of environmental noise at 40 MHz.

4.10.2.3 Received Signal Strength Indicator

The relationship between received signal strength indicator (RSSI) voltage and receiver power was determined by injecting a signal of known strength and measuring the RSSI voltage. This gives the plots shown in Figure 4-41. The plot for each node has a similar gradient of approximately 20 mV/dB in the linear region. The lower left of each plot shows where the transceiver is only detecting noise and is related to the noise figure and hence sensitivity. These tests were conducted at 25°C. During the field tests (Chapter 6) the nodes were subjected to temperatures of 25°C - 35°C.

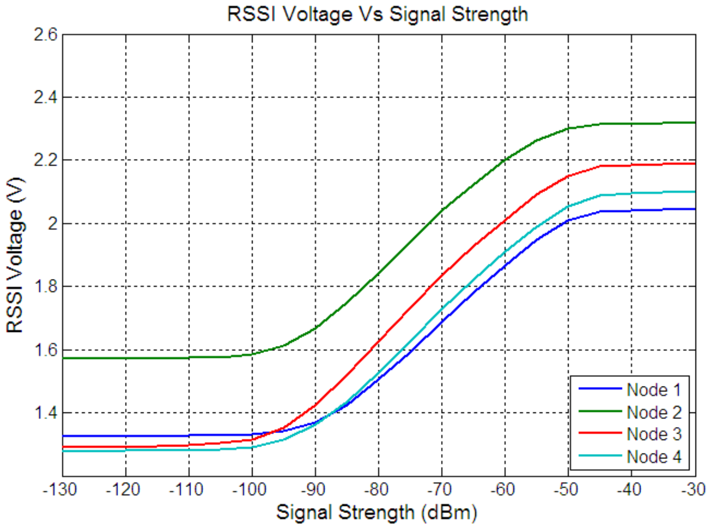


Figure 4-41: RSSI Voltage Vs Signal Strength

It should be noted that the manufacturer specified in [73] that variation in the RSSI voltage Vs RF signal level is normal, but the gradient of curves is relatively constant. Therefore, an adaptive squelch algorithm was used in the nodes and is explained in section 5.3.5.4. It is expected that the RSSI voltage would vary with temperature, due to the receiver thermal noise (25). The adaptive squelch control allows for these variations.

4.11 JCU FLEXIMOTE

A second node, called the JCU Fleximote was developed at a later stage of the investigation. The need for nodes that can communicate using existing radio modems was identified as a result of conference feedback. This allows the node to be used with any commercial radio, thus greatly increasing the range of applications. For agricultural uses, CB radios can be used to greatly extend the range. A typical example application for this technology is an emergency services network, where nodes are connected to radios operating in a licensed frequency band. The use of licensed frequency bands will allow nodes to output higher power levels and achieve longer range communications. The nodes being able to talk to each other will ensure that all personnel involved in fighting a bush fire remain in contact with the base station at all times. This is not the case at present and results in a significant number of deaths each year amongst firefighters.

An example radio modem is the Crescendo UHF/VHF half duplex radio by RF Innovations [74] which operates at selectable frequencies between 148-174 MHz or 380-520 MHz and outputs power up to 5 W with a datarate of 9.6 kbps or 19.2 kbps. This device uses RS-232 for communication with connected devices such as a node.

The JCU Fleximote has further applications when used in conjunction with HF radios. In underdeveloped countries this system could be used by medical aid agencies and alike. By being capable of multi-hop transmissions, the reliable communication range can be extended significantly, particularly since the operating range of HF radios depends on daily variations of the characteristics of the ionosphere.

The JCU Fleximote is based on the original JCUMote design which was altered to accommodate the external radio modem. The RF circuitry was replaced with a

MAX3319E low-voltage RS-232 transceiver IC [75]. The schematic and PCB are shown in Appendix H and Appendix I, respectively. Figure 4-42 and Figure 4-43 show the constructed PCB.

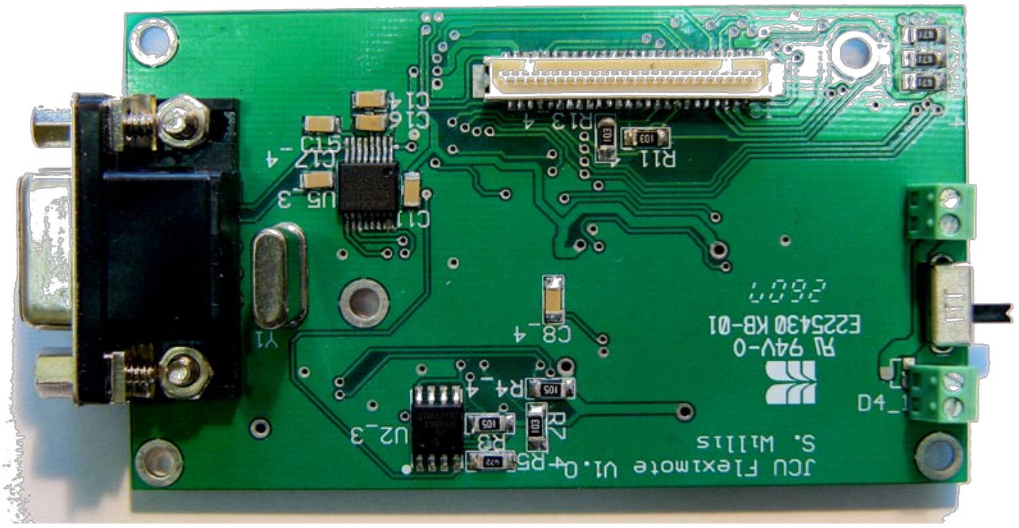


Figure 4-42: Top side of JCU Fleximote PCB

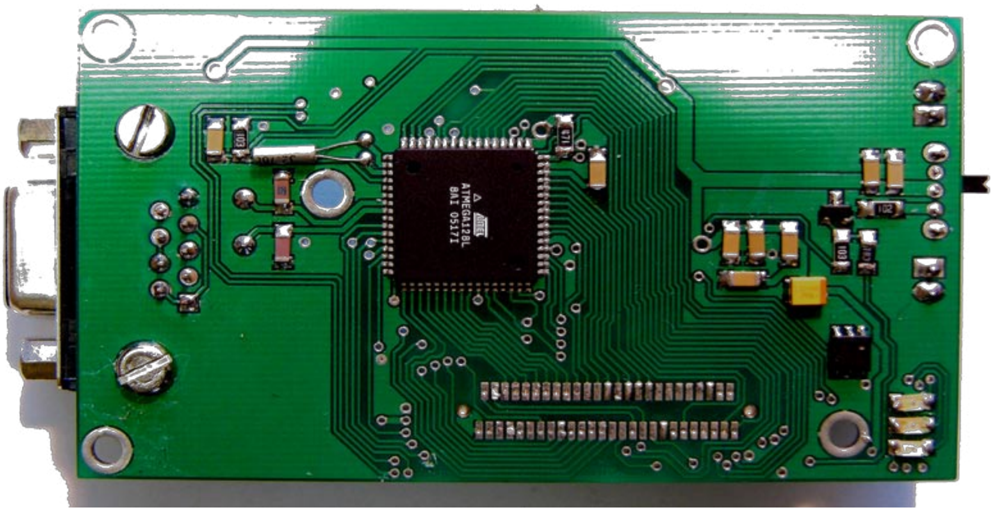


Figure 4-43: Bottom side of JCU Fleximote PCB

4.12 Conclusion

This chapter discussed the complete development of the JCUMote. A novel radio transceiver was designed with external PA and isolation network. Radio antennas were designed and tested. The results of testing showed that the optimal design for the JCUMote is has a quarter-wavelength whip antenna with four radials.

The JCUMote was designed to achieve long-range communications using 1 W of radiated power at 40 MHz. It was shown above that the node transmits the required power level and the receiver gives acceptable results to achieve the goal of long-range communication.

The JCU Fleximote extends the principles of long-range communications by coupling the mote described in this Thesis with commercial radios operating at any frequency.

5 Node Software

An integral component of a Wireless Sensor Network (WSN) is the software that is installed on each node. In a typical WSN, the main task of the software is to take periodic measurements and reliably forward these to the base-station. To achieve this, the software must determine the routes, forward messages and control the radio hardware. In some applications, the software must also perform mathematical operations associated with the processing of the sensor measurements. In addition to these tasks, the software performs node management functions, such as controlling battery usage and coordinating dynamic reprogramming. All software operations must use minimal clock cycles and RAM, because the microprocessor has limited resources.

To coordinate the many activities the node software must perform, several WSN Operating Systems (OS) have been developed. These Operating Systems are packaged with many protocols and applications which reduce the software development time and assist in improving the reliability of the WSN.

This chapter begins with an examination of existing WSN Operating Systems and leads into a further investigation of TinyOS [22], the Operating System selected for use on the JCUMote. Following this is a discussion of the TinyOS implementation on the JCUMote, as well as the software applications that were used in testing.

5.1 Operating Systems for Wireless Sensor Networks

Currently there are four WSN operating systems: TinyOS [22], SOS [76], MANTIS [77] and Contiki [78]. Manjunath [79] analysed these systems in terms of the system architecture, the concurrency model, memory management, network management, dynamic reprogramming service, power management, simulation support and supported hardware platforms.

A WSN operating system (OS) is very different from a traditional computer operating system, because sensor nodes are severely resource constrained. The nodes usually have an 8-bit microprocessor with a 4 to 16 MHz clock speed, code memory in the order of 100 kB and less than 10 kB or RAM. A WSN operating system may be used on a variety of platforms for a wide range of applications. The software is generally event-driven, because nodes react to changes in the environment. The OS must allow tasks to be performed concurrently such as sensor acquisition and data processing.

5.1.1 TINYOS

TinyOS applications are made up of modules called components which are written in NesC. NesC has a C-like syntax and is designed for component-based applications. An application 'wires' required components together to complete the required tasks. TinyOS uses an event-driven concurrency model with two types of processes: Tasks or Events. Events are executed immediately upon request and Tasks have a deferred execution that is controlled by the scheduler. Communication between processes is achieved using a signalling mechanism or by sharing global variables. Access to the shared variables can be protected by placing critical code in atomic sections which disable interrupts to ensure that only one process can access the shared data at a time. TinyOS also allows applications to request dynamic memory using the TinyAlloc memory management component.

Network management in TinyOS is achieved using a light-weight Active Messenger system which specifies message types and distributes messages to the relevant components. TinyOS includes a number of power management systems which put the microprocessor, and other components, into a sleep state if necessary, and adjust the microprocessor state to minimise current draw. Dynamic reprogramming is provided and implemented as a two-step process. Firstly, the software image is received by radio and loaded into the flash memory. Once fully loaded, the node performs a reboot and copies the image into the program memory. Simulation support is provided by TOSSIM which allows developers to compile and simulate TinyOS code on a PC.

TOSSIM is mainly designed for node simulation and has very basic radio propagation models.

A list of hardware platforms supported by each OS is contained in Table 5-1.

TABLE 5-1: SUPPORTED HARDWARE PLATFORMS [79]

TinyOS	SOS	MANTIS	Contiki
Mica	Mica2	Mica2	MSP430 MCU
Mica2	Micaz	Mica2dot	8-bit AVR MCU
Micaz	XYZ Mote	Micaz	
Mica2dot			
Cricket			
TelosA			
TelosB			
Btnode3			
Eyes			
Intelmote			
Intelmote2			
Cricket			
Tmote Sky			
TinyNode 584			

5.1.2 SOS

SOS is also a module based OS. Its software applications consist of binary modules and a kernel. The SOS concurrency model is similar to TinyOS, except that two queues are provided for the processes awaiting execution. The high priority queue is served before the regular queue. SOS modules use messaging and function interfaces for communication between processes. Macros are used to provide the same functionality as the TinyOS atomic sections. SOS also provides dynamic memory allocation and allows modules to request memory blocks that are one of three fixed sizes. The memory management in SOS constantly monitors memory usage and is able to conduct a post-crash memory analysis.

The network management in SOS is similar to the TinyOS active messages. SOS network messages can be addressed to a particular module on a given node. A power management service is also provided and is similar to that of TinyOS. SOS has a reprogramming service that allows individual module binaries to be reloaded instead of a complete image like in TinyOS. Energy is saved on a SOS node, because external flash memory is not required and the node does not need to reboot to load a new module. Similarly to TOSSIM, SOS has a simulator that is able to simulate SOS code on a PC.

5.1.3 MANTIS

Unlike TinyOS and SOS, MANTIS does not have a modular structure. Instead, MANTIS is based on the C programming language and allows existing C code to be ported to the MANTIS platform. MANTIS uses a multithreading concurrency model and implements a time-slicing scheduler. This allows complex tasks to execute in parallel with time sensitive tasks, but with the disadvantage of requiring additional overhead for context switching. Manjunath [79] calculated that this takes less than 1% of the processing time. Inter-process communication is achieved using conventional UNIX inter-process communications mechanisms such as signals, sockets, etc. Data integrity is upheld using a semaphore synchronisation tool, which requires a process to hold a semaphore when it accesses the data. If the semaphore is already occupied, then the requesting process must sleep until the semaphore becomes available. MANTIS developers are not encouraged to dynamically allocate memory.

Network management is implemented using a communication layer which interfaces with the device driver and manages buffering and synchronisation. The MANTIS power management scheme puts the processor to sleep when all threads have declared that they do not require processing time. MANTIS provides an advanced remote management tool that allows users to remotely debug nodes, as well as reflash the OS, reprogram a thread or change variables within a thread. MANTIS programs can be tested on a virtual network on a PC and then deployed into a real network.

5.1.4 CONTIKI

Contiki has a non-modular structure similar to MANTIS. It provides a base system with CPU multiplexing and event handling. Other abstractions are implemented as system libraries that can be linked into a program. A similar concurrency model to TinyOS is used with synchronous and asynchronous events. Multithreading can also be incorporated if an application links in an optional multithreading library. Contiki never disables interrupts and does not allow interrupt handlers to post events, as this can lead to inconsistent shared data. Contiki allows fixed-size memory blocks to be dynamically allocated.

Network management in Contiki is provided by a complete light-weight TCP/IP legacy stack that uses minimal memory. Power management is the responsibility of the software applications which can monitor the list of running processes to determine when the microprocessor can sleep. Similarly to SOS, Contiki allows applications to be dynamically loaded without requiring a reboot. Contiki provides a network simulator that runs on the PC and implements each node as a separate process. It is shown in Table 5-1 that Contiki supports nodes which use Atmel AVR microcontrollers or the Texas Instruments MSP430.

5.2 REVIEW OF TINYOS

TinyOS was selected for the JCUMote, because it is well established, supports numerous platforms and includes many implemented applications and protocols. TinyOS is open-source and is used by thousands of researchers. The TinyOS website usage statistics [80] stated that there were 346125 total downloads over all 17 releases of TinyOS between February 2004 and the 1st June, 2006.

Version 2 of TinyOS was released in 2006 and contains a redesign of many of the core interfaces and abstractions. At the time of release, a large number of TinyOS1 applications and components had not yet been ported to TinyOS2. Therefore, TinyOS1 (version 1.1.15) was selected for the JCUMote.

The TinyOS website contains tutorials [81] that provide a useful introduction to the TinyOS structure. A review of this information is included in Appendix J to aid information provided later in the chapter.

5.3 IMPLEMENTATION OF TINYOS ON THE JCUMOTE

To minimise software development time, the Mica2 platform-specific files in TinyOS were adapted for the JCUMote to account for the hardware differences. To maximise code re-use, the lower layers of the network stack were altered in such a way that the changes were invisible to the upper layers.

The porting of TinyOS to the JCUMote involved rewriting the software interface to the radio transceiver. This included the implementation of a new control interface and Manchester encoder/decoder.

In TinyOS, platform directories are used to hold platform-specific files. A new platform directory was created for the JCUMote and the TinyOS make system was configured to use this directory. All new and modified TinyOS code is contained in Appendix L.

5.3.1 Mica2 Implementation of the Data-Link and Physical Layers

The data-link layer implementation for the Mica2 is defined by a configuration called `CC1000RadioC` (Figure 5-1). This file defines the interfaces to the upper layers, as well as the wiring of the radio transceiver interface to the MAC protocol. The `CC1000RadioIntM` module implements a carrier-sense multiple access with collision avoidance (CSMA/CA) MAC protocol. It provides the `BareSendMsg` interface that is used by the upper layers to send packets and the `ReceiveMsg` interface is used to signal when a packet has been received.

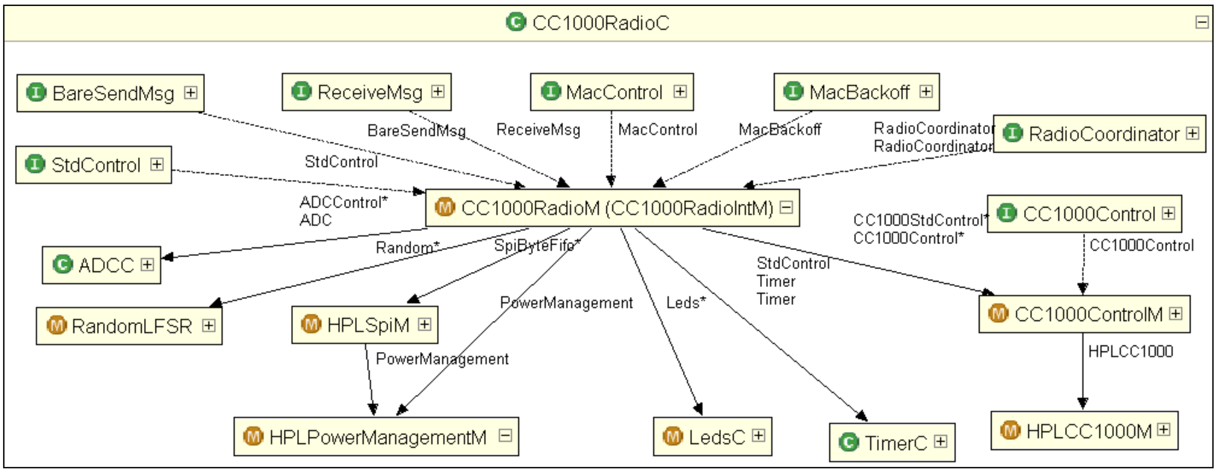


Figure 5-1: CC1000RadioC Component Graph (Component Graphs created using [82])

The `CC1000ControlM` module is used to control the radio transceiver and provides functions to change the mode to transmit or receive, set the carrier frequency, put the transceiver to sleep, etc. `HPLSpiM` provides an interface to the SPI which is used for the transmission and reception of bytes. `ADCC` provides the interface for the analogue to digital converter and it used to measure the received signal strength.

To suit the JCUMote, a new version of `CC1000ControlM` was created to control the TH7122 transceiver IC. The `HPLSpiM` module was replaced with the Manchester encoder/decoder.

5.3.2 JCUMote Radio Transceiver Control

The TH7122 control interface uses a three wire serial protocol that is defined in the datasheet [62]. The TH7122 has four 22-bit control registers which are labelled A to D. A 24 bit word is required to program each register with the first two bits holding the address of the register to be programmed.

A hardware presentation layer module called `HPLTH7122M.nc` was written to handle the control interface of the TH7122. The component graph for the module is shown in Figure 5-2. This module provides the `HPLTH7122` interface for the `TH7122ControlM` module and provides functions to get the status of the lock detect signal, initialise the control lines or write to a control register.

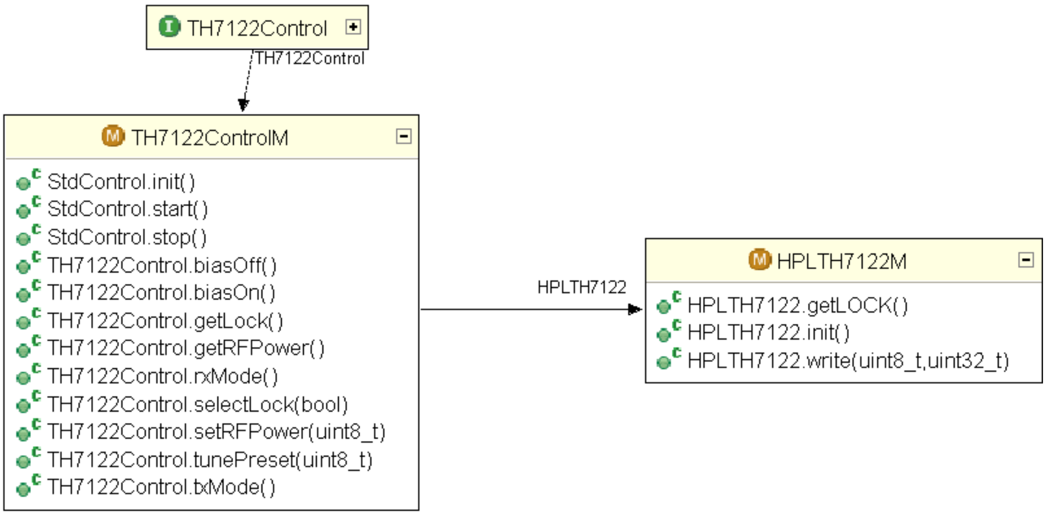


Figure 5-2: TH7122 Control Module

The `TH7122ContolM` module shown in Figure 5-2 provides the `StdControl` interface for starting, stopping and initialising the transceiver. The `StdControl.init()` command sets the phase-locked loop (PLL) to the channel frequency defined in the `TH7122Const.h` file. This file defines the default channel frequency as 40.85 MHz, with alternative settings for 40.75 MHz or 40.9 MHz operation.

The `TH7122ControlM` module is used by the data-link layer protocol to control the transceiver and has the functions shown in Figure 5-2. These functions closely match those of the Mica2 implementation.

Particular attention was given to the `txMode()` command which switches the transceiver to transmit state. It was found that spurious frequency components were generated if the external PA was on while the transceiver was switching to transmit mode. To reduce the problem, the `txMode()` command turns off the internal and external PAs until the PLL is locked (determined by monitoring the lock detect line). Once the PAs have been switched on, the program delays data transmission by 5ms to allow the external PA to reach a steady-state.

5.3.3 MANCHESTER ENCODER

The Manchester encoder and decoder were implemented in the `ManchesterByteFIFOM` module that is shown in Figure 5-3. This module was designed to be functionally the same as the `HPLSpiM` module which was used in the Mica2 for byte level communication with the transceiver. `ManchesterByteFIFOM` provides the `ManchesterByteFIFO` interface which includes the commands shown in Figure 5-3, as well as two events: `dataReady` and `manchesterViolation`. The `dataReady` event is generated when a byte has been transmitted or received. A `manchesterViolation` event occurs when the Manchester decoder detects a Manchester violation, as discussed in section 5.3.4, below.

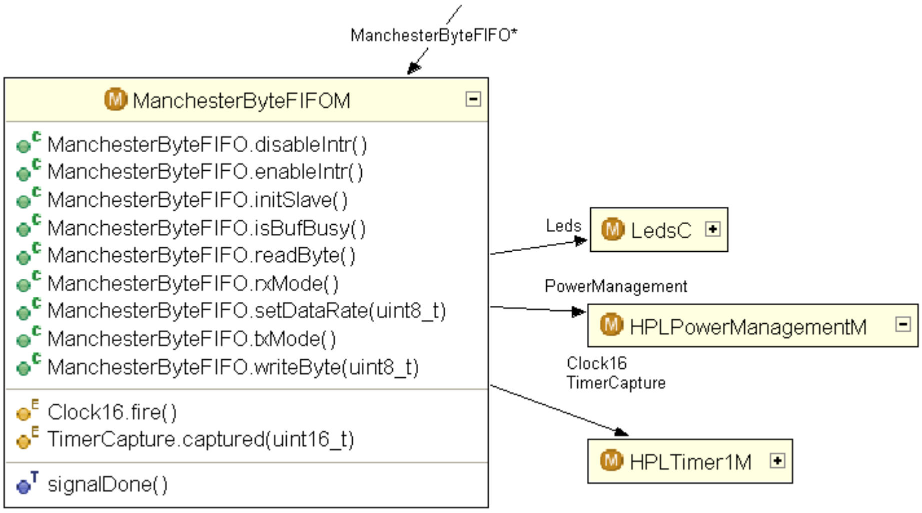

ManchesterByteFIFO*
ManchesterByteFIFOM
ManchesterByteFIFO.disableIntr()
ManchesterByteFIFO.enableIntr()
ManchesterByteFIFO.initSlave()
ManchesterByteFIFO.isBufBusy()
ManchesterByteFIFO.readByte()
ManchesterByteFIFO.rxMode()
ManchesterByteFIFO.setDataRate(uint8_t)
ManchesterByteFIFO.txMode()
ManchesterByteFIFO.writeByte(uint8_t)
Clock16.fire()
TimerCapture.captured(uint16_t)
signalDone()
Leds
LedsC
PowerManagement
HPLPowerManagementM
Clock16
TimerCapture
HPLTimer1M

Figure 5-3: Manchester Encoder Component Graph

The `ManchesterByteFIFOM` module uses the `HPLTimer1M` module which controls the input capture pin used for the Manchester decoder, as well as the output-compare pin, used to generate the SPI clock signal (discussed in section 4.3.6). The power state of the microprocessor is adjusted by `HPLPowerManagementM`.

`ManchesterByteFIFOM` is switched to Manchester encoder mode by calling the `txMode()` command. The SPI and external SPI clock are later activated when interrupts are enabled (by calling the `enableIntr()` command). The `writeByte` command accepts a byte to be transmitted and encodes it into two separate Manchester bytes. Figure 5-4 shows an example of Manchester encoding, where a 0 is represented as a 0 followed by a 1 and a 1 is represented as a 1 followed by a 0.

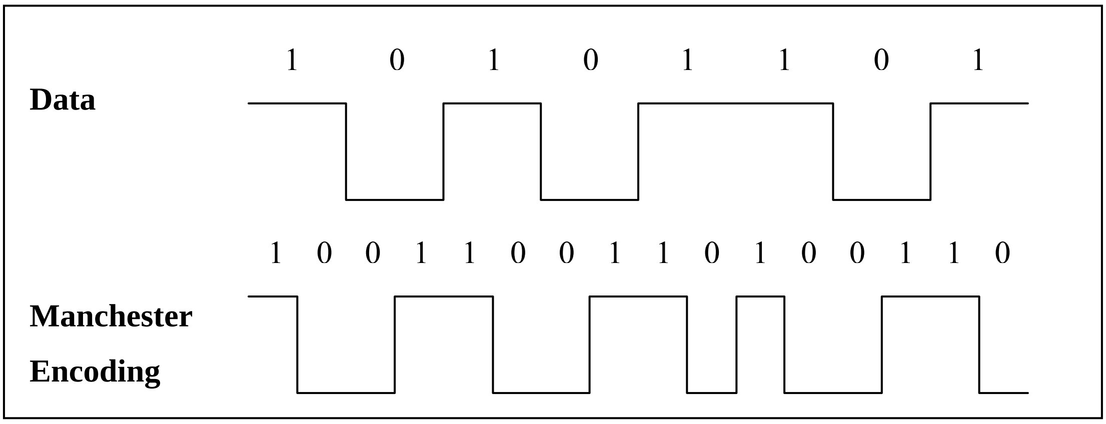


Figure 5-4: Manchester Encoding

To transmit the data, the SPI clock speed is set to twice that of the desired data rate and the encoded bytes are transmitted one after the other. The SPI event-handler coordinates the sending of the second byte and uses the `dataReady` event to signal when both bytes have been transmitted.

A small glitch was discovered when testing the Manchester encoder. The glitch is shown in Figure 5-5 and was found to be caused by the architecture of the SPI. On the ATMega128L, the SPI has a single shift register for transmitting data. An interrupt is generated when this becomes empty and it takes a small period of time before the next byte is loaded. During this time, the shift register contains all zeros and the voltage output on the transmit pin is switched low. In some cases, such as in Figure 5-4, the first Manchester encoded byte ends with a 1 and the next byte starts with a 1. Before loading the second byte, the voltage will momentarily become low and give a waveform as shown in Figure 5-5. Allowances for this glitch were incorporated into the Manchester decoding algorithm.

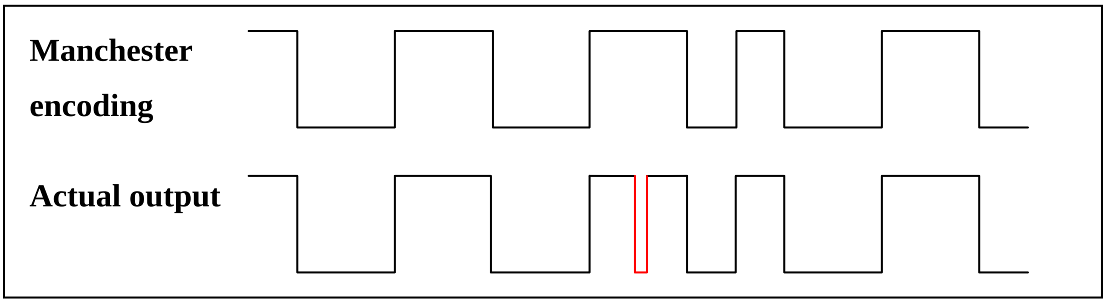


Figure 5-5: SPI Output Glitch

5.3.4 MANCHESTER DECODER

The `ManchesterByteFIFOM` module operates as a Manchester decoder when in receive mode. This mode is activated by calling the `rxMode()` command and enabling interrupts. In this mode, the input capture pin is configured to capture the falling edge of the input waveform. The falling edge is captured instead of the rising edge because the modulation on the transmitter inverts the Manchester encoded waveform. Therefore, capturing the falling edge of the received data is essentially the same as capturing the rising edge of the original data. To avoid confusion, it is assumed in this section that all captures occur on the rising edge.

The decoding algorithm was developed by the author and was later found to be published in an application note by Olmedo of Freescale Semiconductor [83]. The input capture system operates by storing the time of a rising edge and comparing it with the previous measured rising edge.

The transmission time for a Manchester encoded bit is called a chip. The received data can be determined by measuring the number of chips between rising edges and tracking whether the transition is in the middle of a bit or at the end. This is demonstrated in Figure 5-6.

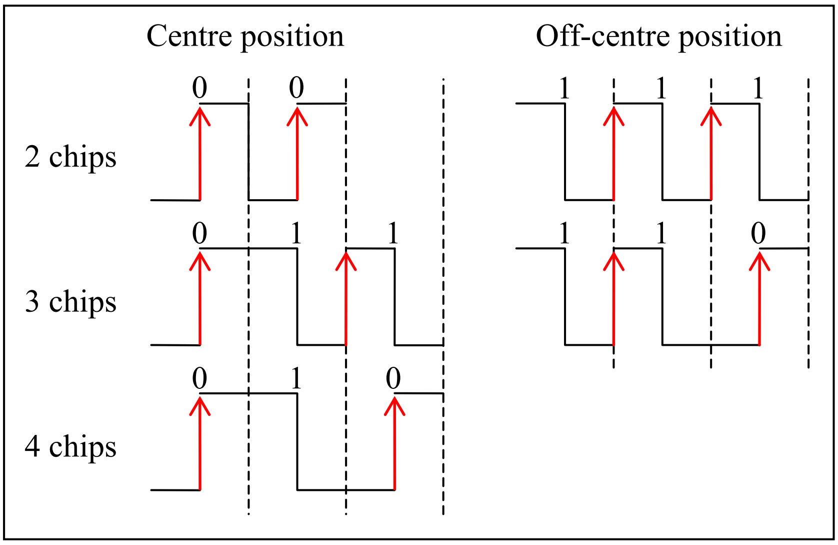


Figure 5-6: Capture of Manchester Encoded Data

The method presented in Figure 5-6 was converted into a truth table (Table 5-2). To decode the data, it is necessary to determine the centre position of the bits being received. This is achieved by sending a preamble which is 10101010. When this is received, the Manchester decoder will count four chips which indicates that the last transition must have occurred in the centre position.

TABLE 5-2: MANCHESTER DECODER TRUTH TABLE

Previous Position	Number of Chips	Decoded Value	Next Value Expected	New Position
Centre	2	0	-	Centre
Centre	3	1	1	Off-centre
Centre	4	10	-	Centre
Off-centre	2	1	1	Off-centre
Off-centre	3	10	-	Off-centre

This algorithm was implemented in `ManchesterByteFIFOM.nc` and uses variables which hold the centre position and last transition time. The algorithm calculates the number of chips when a new transition occurs. The number of clock cycles for a chip is determined by the data rate, which is set in the `TH7122Const.h` header file. To allow for the SPI output glitch, (Figure 5-5) the algorithm ignores the transition if it occurs less than two chips after the previous transition. If the transition time is less than one chip, then it is likely that the transceiver is only receiving noise. In this case, `ManchesterByteFIFOM` disables the input capture interrupts to avoid wasting processor time decoding noise. The program sets a timer which fires after the time equivalent to the reception of one byte. When the timer fires, the decoder is reinitialised and a dataReady event is generated, allowing the TH7122RadioIntM module to perform the routine operations contained in the dataReady event handler.

The state of the decoder is stored in a variable called decState, which is initially set to INIT. Once the first transition has been detected, the state is advanced to the GETLOCK state and the decoder waits to detect the preamble. Once eight preamble bits have been detected, the decoder advances to the DETECT state. If an invalid number of chips are detected at any time, then a Manchester violation is signalled and the decoder reverts back to the IDLE state.

When eight bits have been decoded, the dataReady event is signalled to the TH7122RadioIntM module. The dataReady function in TH7122RadioIntM is fairly complex and is explained in section 5.3.5, below. This function may take considerable time to execute, during which time a new received bit may be missed by the Manchester decoder. To alleviate this problem, the decoder uses two buffers to hold the received data. When the first buffer is filled, the decoder switches to the other buffer and posts a task called signalDone. This allows the decoder to finish execution and wait for the next bit. The signalDone task is executed some time later and signals the dataReady event with the received byte.

The Manchester decoder was tested at data rates up to 19.2 kbps with no overrun problems occurring.

5.3.5 Data-Link Layer

The Data-Link layer is implemented in the TH7122RadioIntM.nc file. This module is an alteration of CC1000RadioIntM which was changed to use the TH7122ControlM and ManchesterByteFIFOM components. The other major difference involves the usage of the received signal strength indicator (RSSI), which is used as a carrier detect mechanism. On the CC1000, the RSSI is inversely proportional to the received power, but on the TH7122 it is directly proportional to the received power. Small changes were made to the carrier-detect and squelch code to allow for this. An additional interface called RadioMonitor was added to the module, which returns the current squelch level to allow upper layer applications to monitor the noise level.

TH7122RadioIntM is entirely event-driven and uses the ManchesterByteFIFOM.dataReady event to conduct operations once a byte has been sent or received. Additionally, timers are used to periodically operate the adaptive squelch control and the low power listening mode which puts the radio in standby mode and periodically wakes it to listen for packets. The period of the timers can be set in TH7122Const.h. TH7122RadioIntM uses two main state variables: RadioState and RadioTxState, which define the current radio state (idle, transmit, receive, etc.) and the current position in the transmit process (preamble, data, CRC, etc.), respectively.

The Data-Link layer uses a packet format defined by the Tos_MSG data type (Table 5-3). The type field contains the active message ID, discussed in section 5.1.1. The group field defines the group of nodes who may receive the message. Nodes that are not part of the group ignore the packet, allowing two separate networks to operate on the same frequency.

TABLE 5-3: TINYOS PACKET FORMAT

Type	Name	Description	Transmitted
uint16_t	addr	Address of packet	Y
uint8_t	type	Type of packet	Y
uint8_t	group	Group address	Y
uint8_t	length	Length of the data part	Y
uint8_t array	data	Main data part	Y
uint16_t	crc	CRC	Y
uint16_t	strength	Signal strength	N
uint8_t	ack	Not used	N
uint8_t	Time	Not used	N
uint8_t	sendSecurityMode	Not used	N
uint8_t	receiveSecurityMode	Not used	N

5.3.5.1 SENDING PACKETS

The BareSendMsg interface is provided for the transmission of packets. To minimise packet collisions (two nodes transmitting at the same time) an initial random packet delay is generated using the MacBackoff interface. The radio is switched to receive mode, the state is set to IDLE and the squelch timer is activated to update the squelch value.

Each time the `dataReady` event fires, the program checks to see if a preamble has been received. If the preamble is not detected after a certain number of `dataReady` events (determined by packet delay time) then the state is advanced to `PRETX` and the analogue-to-digital converter is called to sample the RSSI level. If the RSSI value is less than the squelch value then there must be no other carriers present, so the program switches to transmit mode and sends the preamble. If another carrier was detected, then a new backoff period would be generated using the `MacBackoff` interface.

Once the transmission of data has started, the `dataReady` event fires after each byte has been sent. The program then sends a preamble (length is set in `TH7122Const.h`), a sync byte, the data and then its CRC value. If acks (acknowledgements) are enabled (using the `MacControl` interface), then the transceiver is switched back to receive mode and the program listens for an ack to be sent from the receiving node. Once a packet has been sent and an ack has been received (if enabled), the sending component is notified using a `send.sendDone` signal, which is generated by a task.

5.3.5.2 Receiving Packets

The reception of packets is performed by the `dataReady` event handler, which waits for a preamble. Once a preamble of valid length (set in `TH7122Const.h`) has been received, the algorithm moves to the `SYNC` state. In the sync state, the data is moved through a shift buffer to determine if it holds the SYNC word (0x33CC). The sync algorithm determines the bit offset of the sync byte so that the correct bit shift can be applied to each received byte to reassemble the data into the correct bytes.

Once the sync byte has been detected, the software receives the data field and computes the running CRC. The location of the packet's CRC value is determined from the length field in the packet. Once the CRC value has been retrieved, it is compared with the running CRC value to determine if the packet data is corrupt. If so, the CRC field of the packet is set to 0 to notify the upper layer protocol.

If acks are enabled, the algorithm will send an ack byte preceded by a preamble after the successful reception of a packet. Following this, the current RSSI value is saved into the strength field to notify the upper layers of the signal strength. Once this has occurred, the upper layer software module is notified using a `Receive.receive` signal, which is generated by a task. The `Receive.receive` signal provides a pointer to the receive buffer. The upper layer provides a pointer to a new receive buffer as the return value to the signal.

5.3.5.3 LOW POWER LISTENING

In low power listening mode, the radio is set to standby mode and periodically woken to determine if another node is transmitting. The duty cycle of the radio is determined by a variable called `lplpower`, which is an index to tables in the `TH7122Const.h` file which define the parameters required for radio duty cycles of 100%, 35.5%, 11.5%, 7.53%, 5.61%, 2.22% and 1.00%. `TH7122RadioIntM` provides a command to allow other components to set the low listening mode. By default, this value is 0 which means that the radio is on all the time. This mode was used on the JCUMote.

Low power listening uses a wakeup timer to cycle the radio between the POWER_DOWN and IDLE states. The length of time that the radio is asleep and awake is determined by the tables in the `TH7122Const.h` file. If a preamble is detected while the node is awake, then the radio state is changed to SYNC_STATE and the wakeup timer is set so that it does not fire again until the packet has been received.

If a low-power listening mode is enabled, then the node must transmit a longer preamble so that the receiver will wake up sometime during the preamble to detect it. This means that the transmitter is on longer when low-power listening is used. This method is not very efficient on nodes such as the JCUMote where the current draw is much greater in transmit mode than receive mode.

5.3.5.4 ADAPTIVE SQUELCH

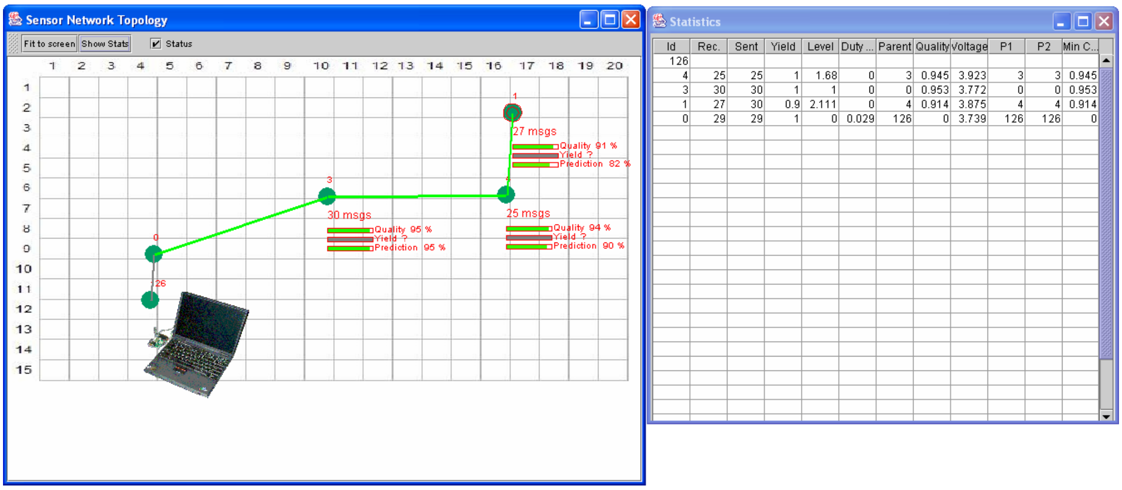
The squelch value is used to detect carrier signals before transmitting. An adaptive squelch mechanism is included, which continuously samples the RSSI value to determine the squelch level. The adaptive squelch mechanism uses a timer to schedule the sampling of the RSSI value. When the RSSI has been sampled, it is placed in a table of nine elements. A task is then posted to determine the squelch level from the values in the table. The new squelch value is calculated by using the third smallest value (`min_value`) in the table to determine the rolling average calculated by equation (31).

$$squelchval = \frac{32 \times squelchval + 2 \times min\_value}{34} \qquad (31)$$

The squelch timer has two interval values defined in `TH7122Const.h`. When the radio first wakes up, a fast interval is used and it is later switched to the slow interval.

5.3.6 Network Layer

The network layer was reviewed in section 2.2.3 and a number of routing protocols were discussed. Typically, a sensor network operates a many-to-one scenario, where all nodes forward their data to a sink node, which is connected to a PC. Several routing algorithms have been implemented in TinyOS, of which, Mintroute was selected. Mintroute is documented in [84] and was selected, because it is platform independent and can be used with a graphical user interface called Surge-View. Surge-View, shown in Figure 5-7, is a free program provided by Crossbow [13] that shows the network topology and the quality of the network links.



Id	Rec.	Sent	Yield	Level	Duty ...	Parent	Quality	Voltage	P1	P2	Min C...
126											
4	25	25	1	1.68	0	3	0.945	3.923	3	3	0.945
3	30	30	1	1	0	0	0.953	3.772	0	0	0.953
1	27	30	0.9	2.111	0	4	0.914	3.875	4	4	0.914
0	29	29	1	0	0.029	126	0	3.739	126	126	0

Figure 5-7: Surge-View Application [13]

To allow packets to reach the source, the routing protocol must implement a route selection mechanism. This is achieved in Mintroute by selecting a suitable parent node to send packets to. The neighbouring node with the lowest estimated route cost is chosen to be the parent. Mintroute maintains a neighbour table which contains a measurement of the reception (inbound) link quality, send (outbound) link quality, parent address and sequence number of neighbouring nodes. This information is used to estimate the link quality and is updated by routing update messages and by snooping messages that are transmitted by the neighbours.

The protocol is implemented using the framework shown in Figure 5-8. The parent selection component uses the neighbour table, which is updated by the estimator. The estimator snoops all of the received messages and estimates the link quality. The table management component uses an algorithm to determine if a node should be removed from the table to allow a new node to be added. The cycle detection component signals to the parent selector if a cycle has been detected. In which case, the parent selector will drop the current parent and choose a new parent from the neighbour table. The parent selection component is also activated by a timer which causes the component to periodically assess whether the current parent is still the optimum. The parent selector is also responsible for sending a route update message to the neighbours. These messages are placed in the originating queue with the application messages. The transmission of these messages is given priority over the forwarding messages which are contained in the forward queue.

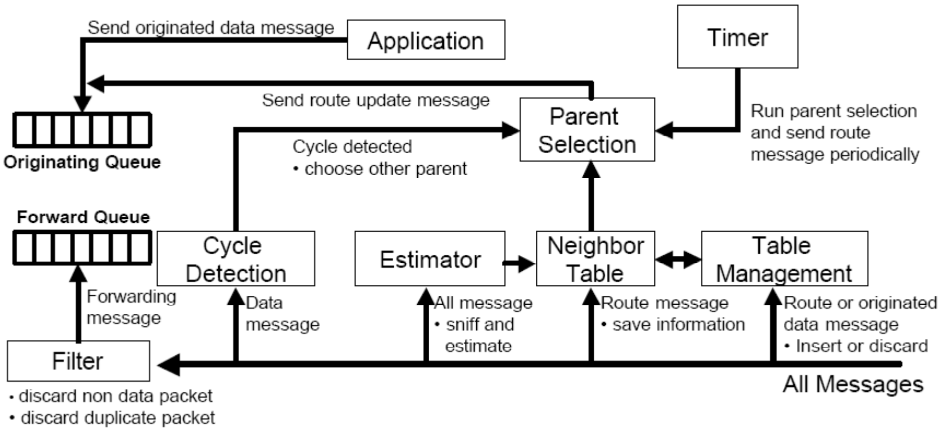


Figure 5-8: Message flow chart showing the routing components [84]

5.3.6.1 LINK ESTIMATION

The quality of a link to a neighbour is determined using the Mintroute estimator, which applies a window mean with exponentially weighted moving average (WMEWMA). The algorithm determines the average success rate during a time period *t*, as shown by equation (32). The average success rate is smoothed using the exponentially weighted moving average (EWMA) function. An estimation of route quality is made after five route messages.

$$success\ rate = \frac{Packets\ received\ in\ time\ t}{max(Packets\ expected\ in\ t, packets\ received\ in\ t)} \qquad (32)$$

5.3.6.2 Parent selection

The parent is selected from the neighbourhood table using a distance-vector algorithm based on the Minimum Transmission (MT) metric. It is assumed that the network uses acks and that poor quality links require retransmissions. Therefore, if the algorithm uses a poor quality route, then the total number of transmissions may be larger than the hop count. For each link, the MT cost is calculated using equation (33) which considers the forward link quality as well as the backward link quality. This reduces the use of asymmetric links, as these have higher cost. The backward link quality is determined from the data provided in the neighbour's route update messages.

$$MT\ cost = \frac{1}{forward\ link\ quality} \times \frac{1}{backward\ link\ quality} \qquad (33)$$

The MT cost is calculated over all hops to the sink node to determine the overall route cost. The MT cost is reported by parent nodes to the children using the periodic routing update messages which are sent every 20 seconds.

The parent selection component will switch to a new parent if: the sink becomes unreachable through that parent, a cycle is detected or a lower cost node is detected. A new node will only be selected as a parent if its MT cost is a pre-defined margin (0.75) below the current cost. If connectivity to the parent is lost and no potential parents are available, then the node declares that it has no parent and disjoins from the tree by setting the routing cost to infinity.

5.3.6.3 Packet snooping

Mintroute uses a `snoop` interface to monitor all packets that are transmitted by the neighbouring nodes. Snooping allows a node to quickly learn about its children and detect cycles. The snooped messages also allow a node to establish who a neighbour's parent is so that an alternative route to the sink may be determined. If a node has no reachable parent, but receives a message to be forwarded, then it rebroadcasts the message with a 'NO_ROUTE' address. All neighbouring nodes will hear the message and quickly learn of an unreachable route.

5.3.6.4 CYCLES

Cycles are detected when a node originates a message and sees it return. This detection scheme works provided the queue management component does not allow forwarded messages to overrun the originated messages. When a cycle is detected, the parent selection component is notified so that a new parent can be selected.

5.3.6.5 DUPLICATE PACKET ELIMINATION

If acks are used, then it is possible that a packet will be retransmitted if the ack is not received. In some cases, the original message will be forwarded as well as the retransmitted message, generating excess traffic. To avoid this problem, an incremental sequence number is attached to each packet. When a packet is received, the most recent sequence number is stored in the neighbour table. If a packet is received with the same sequence number, it is ignored.

5.3.6.6 IMPLEMENTATION

Mintroute is implemented in two main components: `MultiHopEngineM.nc` and `MultiHopEWMA.nc`, which are wired together in `EWMAMultiHopRouter.nc`. `MultiHopEngineM.nc` provides the overall packet movement logic for the multi-hop functionality. This module provides the `Snoop` interface which is used by `MultiHopEWMA` and other modules that wish to examine traffic that is not to be forwarded. The `Intercept` interface is similar to `Snoop`, but only shows packets that are to be forwarded. All incoming packets are forwarded to `MultiHopEngineM` by the Active Message component, `GenericCommPromiscuous`. For traffic forwarding, `MultiHopEngineM` maintains a forwarding packet buffer and uses the `RouteControl` interface provided by `MultiHopEWMA` to determine the parent node. Packets are sent using the `QueuedSend` component.

`MultiHopEWMA` maintains the neighbour table and performs link estimation by snooping packets, as well as directly receiving route update packets that are distributed by the Active Message component, `GenericCommPromiscous`. `MultiHopEWMA` provides the `RouteControl` component that is used by

`MultiHopEngineM` and other components to determine the parent node. Route update messages are scheduled using `TimerC` and periodically sent via the `QueuedSend` component. By separating the multi-hop engine from the link estimation and parent selection component allows developers to easily interchange components.

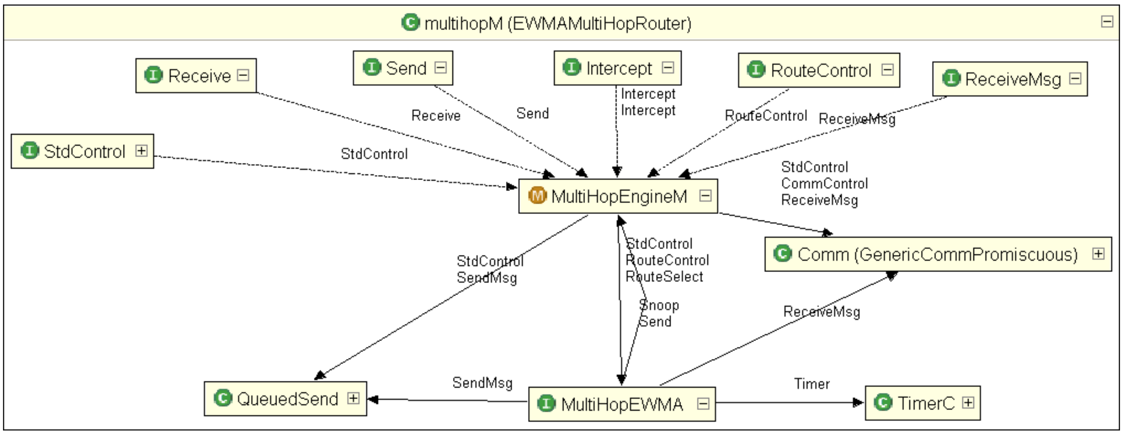


Figure 5-9: Mintroute Component Graph

5.4 Test Programs for the Long-Range Sensor Network

Several programs are discussed below that were developed for field testing or to aid the bench-top testing of the JCUMote. These programs are contained in the `tinyos-1.x/contrib/jcu/apps` directory. There are also several applications that are packaged in the `tinyos-1.x/apps` directory and are useful for hardware testing.

5.4.1 Hardware Verification Software

To test the UART and the onboard flash memory, a simple program called `ReverseUARTEEPROM` was written. A Java program is used to forward a string to the node over the UART. The node then copies the string into the onboard flash memory, retrieves the string and reverses it. The reversed string is sent back to the PC using the UART. This program is based on the `ReverseUART` program contained in the `tinyos-1.x/apps` directory, but has been modified to include the flash memory component.

Another program called `MicaHWVerify-JCU` was used to test the onboard serial ID integrated circuit, the flash memory and the UART. The program is based on the

Mica2 version that is contained in `tinyos-1.x/apps/MicaHWVerify` and has been adapted for the JCUMote so that it uses the UART for communications instead of the radio as is the case with the Mica2. The program tests the flash memory, then captures the serial ID number and forwards this over the UART to the PC to be displayed.

5.4.2 Transmitter Testing Software

A test program was written, called `TxTest2` which allows transmitter hardware to be tested with a constant bit stream. The software interacts with the hardware layer directly and Manchester encodes bytes before they are transmitted. This program allows the frequency deviation to be set using a signal analyser such as the Rohde and Schwarz FSIQ.

5.4.3 Receiver Testing Software

A receiver test program was written, called `RxTest2` which accepts the Manchester decoded bytes from the receiver and transmits the bytes over the serial port. The program was used in conjunction with `TxTest2` to evaluate the Manchester encoding and decoding process. The program was also used with a vector signal generator (Rohde and Schwarz SMIQ) and an I/Q modulation generator (Rohde and Schwarz AMIQ) to test and tune the receiver sensitivity and bit error rate performance.

5.4.4 Transmitter-Receiver Switch Testing Software

To test the start-up performance of the transmitter, a transmit-receive switching program called `TxRxSwitch` was written. The program switches between transmit and receive mode twice a second. This allows the switching time to be determined and also allows the transmitter start-up sequence to be tuned in order to minimise spurious frequency components.

5.4.5 Radio Transmission Range Testing Software

To test the radio transmission range, a node was programmed with `CntToLedsAndRfm`, which is distributed with TinyOS in the `tinyos-1.x/apps/CntToLedsAndRfm` directory. The program increments a counter four times a second, displays the value on the LEDs and transmits the counter value in a packet.

Upon reception, the receiver program, `RfmToLeds`, displays the counter value on the LEDs. These programs are useful for the bench-top testing of the radio hardware. For signal strength testing, `RfmToLeds` was extended and named `SignalStrengthTest`, shown in Figure 5-10.

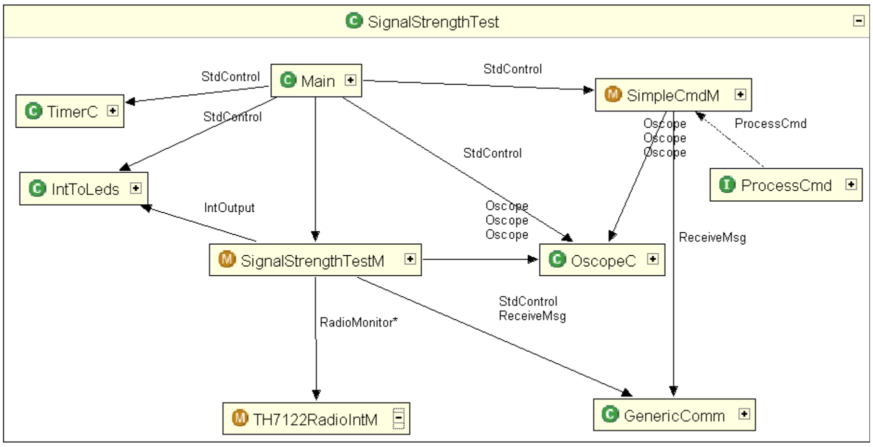
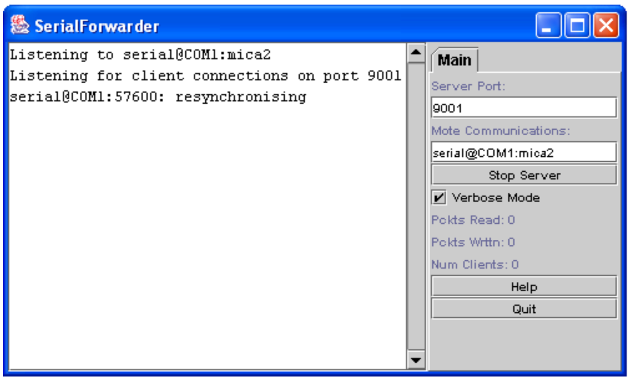


Figure 5-10: Component diagram for signal strength test program

Packets are received through the `GenericComm` component and the counter messages are distributed to `SignalStrengthTestM`, which uses the counter value to determine the packet loss. Upon reception, the signal strength is retrieved from the packet and the noise is measured using the `RadioMonitor` interface of `TH7122RadioIntM`. The readings of signal, noise and packet loss are sent to the `OscopeC` package which buffers ten readings and then forwards them over the COM port. Functionality was added to allow the user to view the measurements before all ten values have been received in the buffer. To perform this task, an existing TinyOS application (BcastInject) was modified to allow the user to issue a dump buffer command (“`java net.tinyos.tools.BcastInject led_on`”). The `led_on` command simply represents command number 1. This java program places the number 1 into a `SimpleCmd` message and sends it to the node using the UART. Upon reception, the `GenericComm` command distributes the message to the `SimpleCmd` component, which determines that the command type is ‘1’ and sends a run of 0s to the `OscopeC` component. This flushes the `OscopeC` buffer which sends the existing buffered data over the UART.

On the PC, a program called SerialForwarder (Figure 5-11) accepts messages from the COM port and forwards them to TCP port 9001. Many of the java applications packaged with TinyOS (such as BcastInject and Surge-View) use SerialForwarder to send and receive packets from the base station node.


```
Listening to serial@COM1:mica2
Listening for client connections on port 9001
serial@COM1:57600: resynchronising
```

Server Port: 9001

Mote Communications: serial@COM1:mica2

☑ Verbose Mode

Pckts Read: 0

Pckts Wrttn: 0

Num Clients: 0

Figure 5-11: SerialForwarder Application

SerialForwarder forwards the `Oscope` messages to a package called Oscilloscope (Figure 5-12), which retrieves the measurements from the `Oscope` message and graphs them against time.

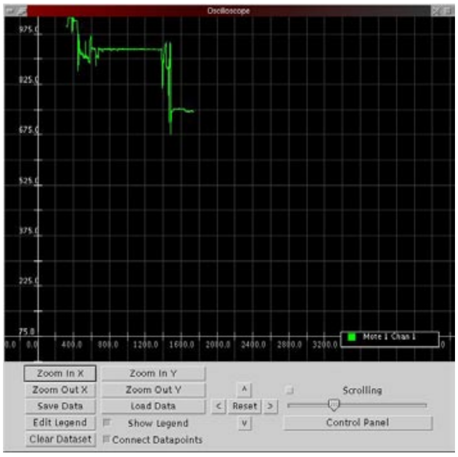


Figure 5-12: Oscilloscope Application

5.4.6 Long-Range Wireless Sensor Network Testing Software

To test the long-range network over a suburban area, an application was developed called LRNet (**L**ong-**R**ange **Net**work). LRNet was designed to emulate the typical sensor network scenario of taking periodic measurements and forwarding them to a base-station. LRNet is based on the Surge application which uses an attached sensor board and forwards measurements using a routing protocol. By making LRNet similar to Surge, the Surge-View package (Figure 5-7) can be used to view the network topology and link reliability. Additionally, the MoteView package (shown below) can be used to log the network data.

In a normal WSN, data is collected from a sensor board. In the LRNet application, the data is the measured performance of the radio links. This allows the signal strength to be compared with predictions from the radio propagation model presented in Section 3.3. LRNet maintains a neighbourhood table that contains the five strongest neighbouring nodes. The last recorded RSSI and noise measurements are stored for each neighbour. Every eight seconds the data for a node is removed from the table in a first-in, first-out fashion and transmitted in a packet with the current battery and solar voltages.

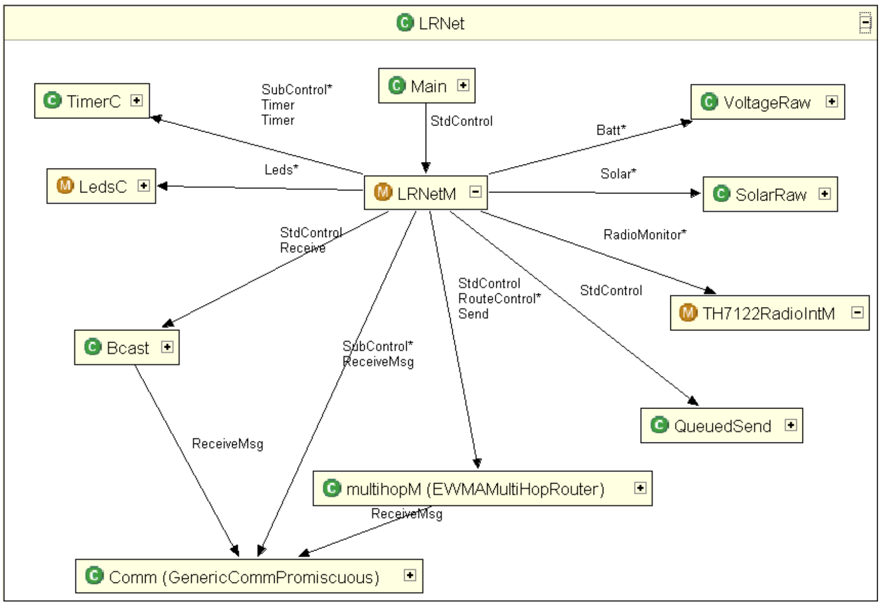


Figure 5-13: LRNet Component Graph

Figure 5-13 shows the component graph for the LRNet application. `LRNetM` updates the neighbour table by monitoring all packets received though the `ReceiveMsg` interface of `GenericCommPromiscuous`. The network table is maintained so that a strong node overwrites a weaker node. Once a neighbour's details have been transmitted, the signal strength is set to zero so that the node does not remain in the table indefinitely.

The measurement sequence is started by a timer which fires every eight seconds and calls the ADC to sample the solar voltage. The ADC event later fires and starts the battery voltage monitor, which, upon completion posts a task called `sendData`. This task places the solar, battery and link performance data into a packet and calls Mintroute (`EWMAMultiHopRouter`) to select the parent. The packet is then transmitted using Mintroute with node 0 (the base station) set as the destination. Each node maintains an LRNet sequence number which is placed in the packets and is incremented each time a message is sent. The sequence number can later be used to determine the number of lost packets.

LRNet also implements the Surge capability of allowing communications from the base station to a node. A particular node can be controlled from the base station using command messages. Mintroute is a many-to-one routing protocol and does not allow messages to be delivered from the sink to a particular node (e.g. One-to-one routing). Therefore, Surge uses a broadcasting scheme (`Bcast`) to accomplish this task. `Bcast` operates by retransmitting any `Bcast` messages that are not addressed to that node. To avoid rebroadcasting messages, `Bcast` stores the most recent sequence number and ignores messages with the same number. The `Bcast` messages are automatically delivered to the `Bcast` component by `GenericCommPromiscuous`. The `Bcast` component forwards messages with a local address to the `LRNetM` module. Several command messages are defined that allow the user to adjust the sampling rate, put a node to sleep, wake a node or focus on a node. When a node is focussed, it increases its sampling rate to one sample per second. In Surge, a sounder board is attached and this is activated when a node is focussed. In LRNet, the node turns on the LEDs when focussed.

In addition to the above features, LRNet also uses a watchdog timer as a safety measure in the prototype network. If a packet is not received for two minutes then the software resets the radio transceiver and routing protocol. Each time a packet is received, the timer is reset.

To allow the use of the Surge-View application (Figure 5-7) the LRNet messages (shown in Figure 5-14) were made the same size and type as the Surge packets. Surge places the battery voltage as the most significant bits in the 32-bit sequence numbers. On the Mica2, the maximum battery voltage reading requires 9 bits, but on the JCUMote 10 bits are required. This means that the JCUMote allows for 22 bit sequence numbers, whereas Surge allows for 23 bit sequence numbers. This will not cause a problem, because the highest 22-bit sequence number is 4194303, which would take 388 days to be reached if a message is transmitted every eight seconds.

	8	16	16	32	8	8	8	8	8	8
Surge	Type	Reading	Parentaddr	Sequence Number	Light	Temp	Magx	Magy	Accelx	Accely

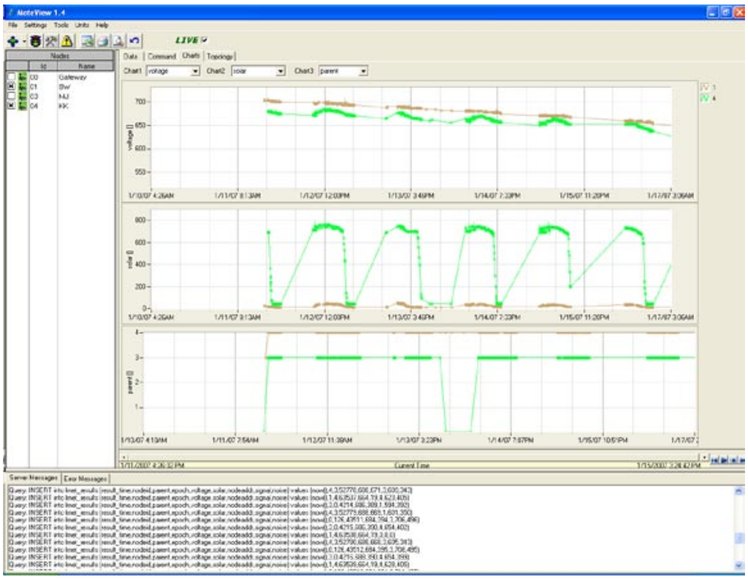
	8	16	16	32	16	16	16
LRNet	Type	Solar	Parentaddr	Sequence Number	Neighbour	Strength	Noise

Figure 5-14: Surge and LRNet packet formats

5.4.6.1 MOTEVIEW

MoteView is a free software package that is provided on the Crossbow website [13] and is designed to allow users to easily deploy and monitor a wireless sensor network. The software is made up of two components for the server and the client. The server software, called Xserve is connected to the gateway node by serial, USB or a network interface and receives the network packets that are forwarded by the node. Xserve extracts the data from the fields of the packet and stores the data in a PostgreSQL database. The client software, Moteview connects to the database to display the network data and topology. Figure 5-15 shows MoteView graphing the battery voltage, solar voltage and parent address for two of the JCUMote nodes.

MoteView and Xserve are primarily designed for the Mica series motes running a Crossbow network protocol called XMesh. Xserve also handles Surge packets and uses xml files that specify how the data is parsed from the network messages and placed into the database. A suitable xml file was created for LRNet and the results are stored in a database table called lrnet_results. The data can be viewed using MoteView or queried using SQL (Structured Query Language).



```
Query: INSERT into lnet_results (result_time,nodeid,parent,epoch,voltage,solar,nodeaddr,signal,noise) values (now(),4,3,52770,606,671,3,630,343)
Query: INSERT into lnet_results (result_time,nodeid,parent,epoch,voltage,solar,nodeaddr,signal,noise) values (now(),1,4,63537,664,19,4,623,405)
Query: INSERT into lnet_results (result_time,nodeid,parent,epoch,voltage,solar,nodeaddr,signal,noise) values (now(),3,0,4214,606,309,1,594,392)
Query: INSERT into lnet_results (result_time,nodeid,parent,epoch,voltage,solar,nodeaddr,signal,noise) values (now(),4,3,52775,608,669,1,631,350)
Query: INSERT into lnet_results (result_time,nodeid,parent,epoch,voltage,solar,nodeaddr,signal,noise) values (now(),0,126,43511,684,394,1,706,496)
Query: INSERT into lnet_results (result_time,nodeid,parent,epoch,voltage,solar,nodeaddr,signal,noise) values (now(),3,0,4215,606,390,4,654,402)
Query: INSERT into lnet_results (result_time,nodeid,parent,epoch,voltage,solar,nodeaddr,signal,noise) values (now(),1,4,63538,664,19,3,0,0)
Query: INSERT into lnet_results (result_time,nodeid,parent,epoch,voltage,solar,nodeaddr,signal,noise) values (now(),4,3,52790,606,669,3,636,343)
Query: INSERT into lnet_results (result_time,nodeid,parent,epoch,voltage,solar,nodeaddr,signal,noise) values (now(),0,126,43512,684,395,3,706,495)
Query: INSERT into lnet_results (result_time,nodeid,parent,epoch,voltage,solar,nodeaddr,signal,noise) values (now(),3,0,4215,588,390,4,654,399)
Query: INSERT into lnet_results (result_time,nodeid,parent,epoch,voltage,solar,nodeaddr,signal,noise) values (now(),1,4,63539,664,19,4,629,405)
```

Figure 5-15: MoteView Chart Display

5.4.7 Over the Air Programming

Over the air programming is an important feature to have in a wireless sensor network, particularly when the nodes are separated by large distances. TinyOS includes the Deluge package which performs this task. The software images used by Deluge are stored in the onboard flash memory. When a new image is injected into the network, it is loaded into the flash memory on each node and forwarded on. Once an image is loaded, the node performs a reboot and loads the image from the flash memory into the program memory. A node running Deluge continuously monitors the neighbours to ensure that it has the most recent program image. If a node has an incomplete image in memory, then Deluge will coordinate a neighbouring node to forward the missing parts of the image. Deluge allows for three program images to be stored in the flash memory. Image 0 is called the Golden Image and can only be programmed using the serial port. The Golden Image is a failsafe image that is loaded if the node detects problems with one of the other images or has problems loading a new image.

Deluge includes a java program that is used to inject images into the network, query nodes and request a new image to be loaded. Instructions on using the Deluge software are contained in [85]. To include the Deluge capability in an application, the `StdControl` interface of the `DelugeC` component must be wired to `Main.StdControl`.

When a node is programmed with an application, the TinyOS make system automatically installs the `TOSBoot` component. This component is responsible for loading a new program image when the node starts. Before an image is loaded, `TOSBoot` checks the node voltage. Since the voltage monitoring circuitry is different on the JCUMote to the Mica2, the `TOSBoot` component was modified to suit.

To test Deluge, the nodes were set up on the bench and installed with Deluge as the Golden Image. A simple LED counting program was injected into the network and saved in memory position 1. All nodes received the program and performed a reboot successfully. Following this, a third program was injected into the network successfully. In some cases, one node would not reboot at the same time as the others because it had not yet received the full image. This node would request the remainder of the image from the other nodes and perform a reboot when ready. To test the Golden Image, the boot process was interrupted three times by resetting the power. TOSBoot detected this as a problem and automatically loaded the Golden Image from the flash memory.

Deluge was also tested with the Surge application. The image was successfully propagated into the network and the nodes rebooted. However after a reboot, no packets were received at the base station from the network, even though the nodes were observed to be transmitting. Further investigation is required to find the cause of this problem. It was possible that the nodes were not receiving valid route updates and therefore did not know the route to the sink. It was decided to ignore this problem and not use Deluge with LRNet. The remote programming capability was not required for the purpose of the long-range network field test.

5.5 Conclusion

A review of the TinyOS, SOS, MANTIS and Contiki wireless sensor network operating systems was conducted and TinyOS was selected for use with the JCUMote because it is packaged with many protocols and applications that can improve the software development time. The TinyOS implementation for the Mica2 was modified to suit the JCUMote. The software interface to the radio transceiver chip was written to support the Melexis TH7122 transceiver IC used on the JCUMote. A Manchester encoding and decoding algorithm was developed and implemented in TinyOS for the JCUMote.

The modular structure of TinyOS allowed a number of testing applications for the JCUMote to be developed quickly. The major application developed for testing the operation of a long-range wireless sensor network is LRNet. For this application, the Mintroute network protocol was selected and installed on the JCUMote. The LRNet software was developed to detect the signal strength of the neighbouring nodes and forward the results across the network to the base station, which is connected to a PC running MoteView. MoteView was configured for use with the JCUMote and was used to log the results of the field testing discussed in the next chapter.

6 Results

To achieve the project aim of *investigating what changes are required to existing wireless sensor nodes to achieve long-range communications,* a four-node proof-of-concept long-range wireless sensor network (LRWSN) was installed in a suburban environment. The configuration of the LRWSN is discussed in this chapter and measurements of the system reliability and performance are presented. Additionally, the results of the radio testing are discussed and used to validate the Wireless Sensor Network Radio Propagation model (WSN model) presented in section 3.3. The results discussed in this chapter are presented by Willis and Kikkert in [86] (this paper is included in Appendix D).

6.1 Node Transmission Range Testing

The transmission range of the nodes was determined in suburban and rural environments. The tests were performed by installing a transmitting node at a fixed position and recording the signal strength at various locations using a receiving node connected to a laptop computer.

6.1.1 Methodology

The transmitting node was programmed with the `CntToLedsAndRfm` application (section 5.4.5). This program places a counter value in a packet and transmits packets at a rate of four times a second. The node was installed with a quarter wavelength whip antenna surrounded by four quarter wavelength radials declined at an angle of 30^o to the horizontal, as shown in Figure 6-1.

The receiving node was programmed with the `SignalStrengthTest` application (section 5.4.5). This program receives the counter value and displays the current signal strength, noise and packet loss on a connected laptop computer that is running the Oscilloscope software (section 5.4.5). The receiving node was positioned on the roof of a car and readings were taken at various locations where the GPS position was recorded. Each recording was taken when the car was stationary. At each position a number of readings were taken and the mean of these was recorded.

6.1.2 Suburban Environment Testing

Testing was performed in the Townsville (in Queensland, Australia) suburb of Annandale. The transmitting node (Figure 6-1) was positioned 13 m above ground on the roof of the Electrical and Computer Engineering building at James Cook University in the neighbouring suburb of Douglas.

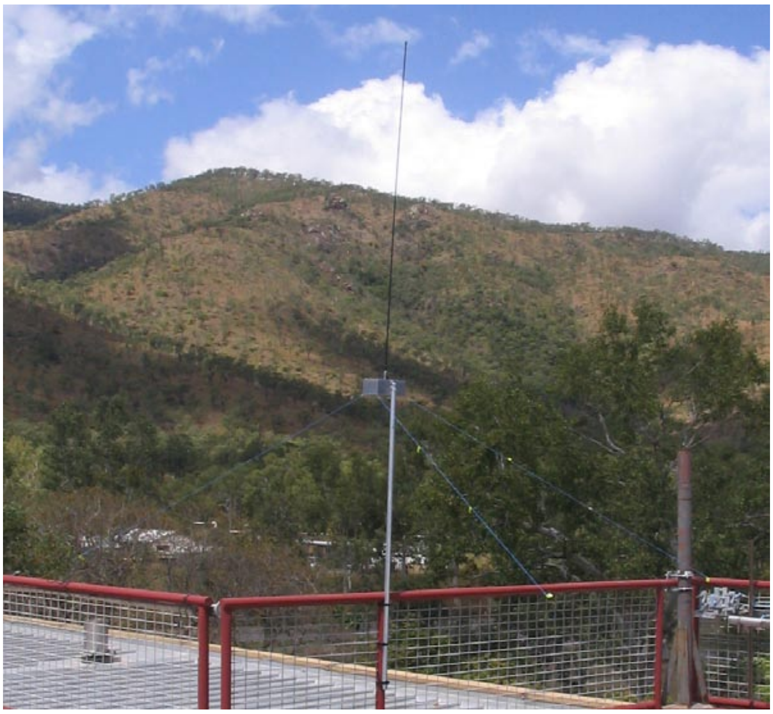


Figure 6-1: Transmitting node installed on roof

The receiving node was placed on the roof of a car approximately 1.5 m high (the radials were not installed). Measurements were taken at the locations shown in Figure 6-2. The transmitter has an output power of +30 dBm and was positioned at the point denoted as ‘JCU’. Signal strength measurements were taken at points 1 to 4, giving the results shown in Table 6-1, which also shows the distance and receiver elevation relative to the transmitter. At position 3, the received signal strength was -81.7 dBm, which is very close to the tested receiver sensitivity limit of -81 dBm (node 2 from section 4.10 was used). No packets were received at position 4 (2.8 km) which suggests that the received signal strength would be below the receiver sensitivity level at this distance.

TABLE 6-1: RESULTS OF SUBURBAN TESTS

Test Position	Latitude	Longitude	Distance (km)	Elevation (m)	Signal Strength (dBm)
Transmitter	19°19'53.59"S	146°45'29.50"E	-	-	-
1	19°18'52.11"S	146°46'11.86"E	2.2	-28	-73.5dBm
2	19°18'50.54"S	146°46'26.27"E	2.5	-30	-74.5dBm
3	19°18'50.46"S	146°46'36.02"E	2.7	-32	-81.7dBm
4	19°18'49.63"S	146°46'39.66"E	2.8	-32	-

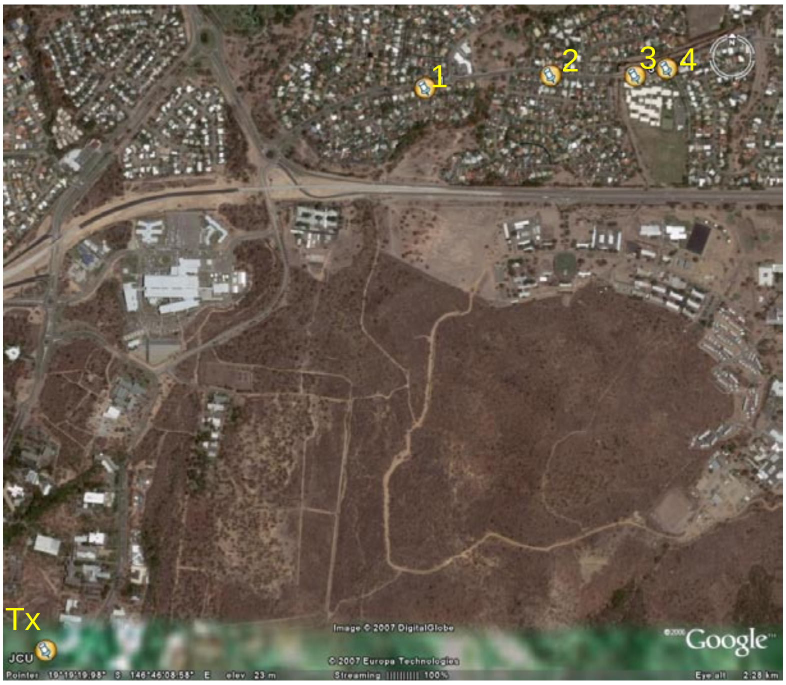

1
2
3
4
Tx
JCU
Image © 2007 DigitalGlobe
©2006 Google
© 2007 Europa Technologies
Pointer 19°19'19.98" S 146°46'08.58" E elev 23 m
Streaming |||||||||| 100%
Eye alt 2.28 km

Figure 6-2: Node Positions for Suburban Tests (Source: DigitalGlobe and Google)

Figure 6-3 shows the terrain profile for the link between the transmitter and point 3. The height of the nodes is represented with the red lines. The terrain profile to the other test positions would be slightly different as the test positions are on a different heading from the transmitter. The transmitter was installed 13 m above ground level on a building whose elevation is 32.8 m above point 3.

It should be noted that buildings and vegetation are not shown on the terrain profile. These will cause the signal to be attenuated and reflected, as discussed in section 3.1.2. The suburban environment is also a source of RF noise, which may reduce the system performance. Additionally, the car roof is not an ideal ground plane and variations in the received signal strength were noted by moving the node around the car roof. Generally the node was positioned on the side furthest from the transmitter to maximise the area of the ground plane between the receiver and the transmitter.

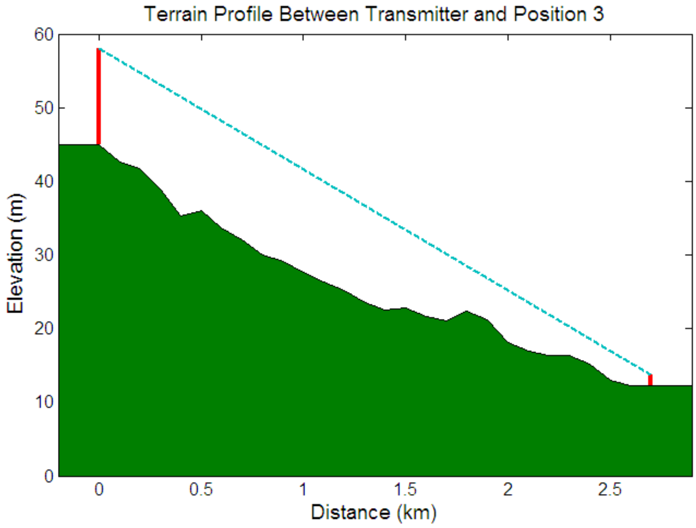


Figure 6-3: Terrain Profile Between Transmitter and Position 3

6.1.2.1 Predicted Results

The Wireless Sensor Network Radio Propagation model (WSN model) presented in section 3.3 was used to give the results shown in Table 6-2, where pR_{meas} is the measured signal strength, pR_{mean} is the mean predicted signal strength, SNR_{mean} is the mean predicted signal-to-noise ratio and σ is the standard deviation of the predictions. The number of reflections is shown by *num. reflections*. Case zero occurs when there is a direct ray only, case one occurs when there is a direct ray and a ground reflection and case two occurs when there is a direct ray, ground reflection and one multipath reflection. To generate these predictions, the WSN model was run for 1000 iterations for each link. The mean SNR, mean received power and standard deviation were calculated over the 1000 values. Predictions generated by the Plane–Earth model (section 3.1.2) are also shown. For these calculations, the receiver height was 1.5 m and the transmitter height was the sum of antenna height (13 m) and the difference in terrain elevation (see Table 6-1).

TABLE 6-2: COMPARISON OF MEASURED AND PREDICTED RESULTS FOR SUBURBAN TESTS

Posn.	pR_{meas} (dBm)	Dist. (km)	Num. reflections	0	1	2	3	4	5	Plane-Earth model
1	-73.5	2.2	pR_{mean} (dBm)	-50.7	-76.9	**-73.7**	-70.2	-68.2	-67.1	-66.4
			SNR_{mean} (dB)	39.8	13.6	16.8	20.3	22.2	23.4	24.1
			σ (dB)	0.00	0.34	7.40	6.60	6.66	6.58	0.00
2	-74.5	2.5	pR_{mean} (dBm)	-52.0	-79.3	**-75.8**	-72.1	-70.1	-68.3	-68.2
			SNR_{mean} (dB)	38.5	11.2	14.7	18.4	20.4	22.2	22.3
			σ (dB)	0.00	0.34	7.77	7.06	6.98	6.16	0.00
3	-81.7	2.7	pR_{mean} (dBm)	-52.7	**-80.7**	-76.0	-72.5	-70.4	-68.9	-69.2
			SNR_{mean} (dB)	37.8	9.8	14.5	18.0	20.0	21.5	21.3
			σ (dB)	0.00	0.34	7.61	7.05	6.70	6.75	0.00
4	-	2.8	pR_{mean} (dBm)	-53.7	-98.1	-77.1	-74.5	-71.6	-69.6	-69.8
			SNR_{mean} (dB)	36.8	-7.61	13.4	16.0	18.9	20.9	20.7
			σ (dB)	0.00	0.87	6.00	7.86	7.26	6.72	0.00

Table 6-2 shows that the mean received power increases as the number of multipath components increase. The best prediction for the received signal strength is shaded in grey and was found to be within 1.7 % (1.3 dB) of the measured value in all cases. For positions 1 and 2 the model gives the most accurate prediction when there is one multipath signal, but in position 3 the model gives the most accurate prediction when no multipath reflections are assumed. Normally, the user of a propagation model would not be expected to know the number of multipath components. Section 6.2.3 presents an empirical model that provides an estimate of the number of components based on the number of buildings in the Fresnel zone and the height of the nodes. This model was developed by the author from the data collected from the field tests.

It is difficult to measure the presence of the multipath components, but the possibility of having a small number of reflections seems feasible. Reflections from specular surfaces are complex, but as a general guide, an object must be much larger than the wavelength, which is 7.5 m at a frequency of 40 MHz. In a suburban environment, a house may be large enough to act as a reflector and generate multipath signals. It should also be noted from Figure 6-2 that positions 1 and 2 are more enclosed by buildings than positions 3 and 4 and may therefore be more prone to multipath reflections. This may explain the differences in model predictions.

Table 6-2 also shows the signal-to-noise (SNR) predictions. For a reliable transmission, a SNR of at least 7.59 dB is required. This was calculated using the counter packet size of 120 bits, including 2 preamble bytes and 2 sync bytes. Therefore, a bit-error-rate less than 8.3×10^{-3} is required to receive an entire packet. Using equation from section 3.6.2, the required SNR can be determined. Table 6-2 shows that in position 3, the mean predicted SNR is above this threshold, but in position 4 the predicted SNR is much less than the threshold and therefore no packets are received.

For comparison with the WSN model, the received signal strength was calculated using the Plane-Earth model presented in section 3.1.2. This model was chosen for comparison because it includes the ground reflection and it was stated by Hernando *et al.* [41] that the model gives reasonable results for low antenna heights. This model gave predictions within 15.3% (12.5 dB) of the measured values which is worse than the WSN model, but may be useful for giving approximations.

6.1.3 Rural Environment Testing

Rural testing was performed at Hervey Range which is approximately 30 km west of the James Cook University campus in Townsville. The transmitter was installed on a 1.8 m star-picket at the top of the mountain range. A quarter wavelength whip antenna was used with four radials, as shown in Figure 6-4. The same transmitting and receiving nodes were used for the rural tests as those used for the suburban tests.

The signal strength was measured using a receiving node positioned on the roof of a car. Periodic measurements were taken at the base of the mountain range at the test positions shown in Figure 6-5. Table 6-3 shows the measured signal strength, as well as the distance and elevation of each test position relative to the transmitter.

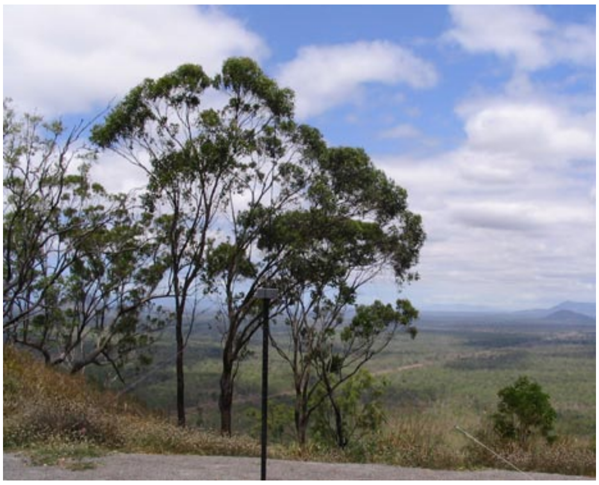


Figure 6-4: Transmitter for Rural Environment Field Tests

TABLE 6-3: RESULTS OF RURAL TESTS

Test Position	Latitude	Longitude	Distance (km)	Rel. Elevation (m)	Signal Strength (dBm)
Transmitter	19°21'20.66"S	146°27'50.82"E	-	-	-
1	19°21'17.52"S	146°29'30.02"E	2.8	-245	-56.3
2	19°20'40.26"S	146°30'4.64"E	4.0	-264	-80.3
3	19°20'8.11"S	146°30'43.70"E	5.5	-275	-67.3
4	19°19'42.32"S	146°31'17.36"E	6.7	-281	-70.4
5	19°19'19.78"S	146°31'54.68"E	8.0	-284	-73.8
6	19°19'15.84"S	146°33'11.55"E	10.1	-292	-77.2
7	19°19'5.90"S	146°33'48.60"E	11.2	-295	-76.3
8	19°18'59.78"S	146°34'21.76"E	12.2	-299	-77.5
9	19°18'56.95"S	146°34'58.94"E	13.2	-300	-84.2

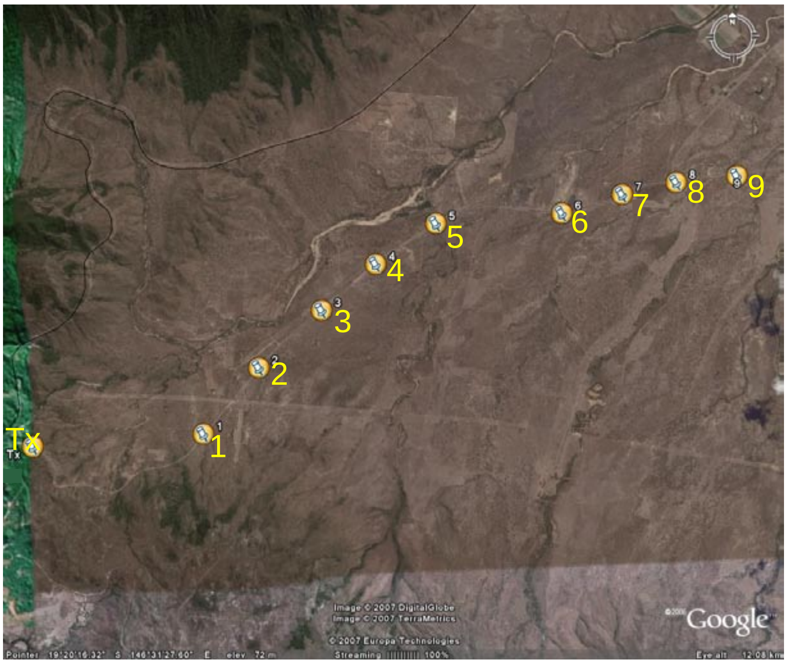

Tx
1
2
3
4
5
6
7
8
9
Image © 2007 DigitalGlobe
Image © 2007 TerraMetrics
© 2007 Europa Technologies
Google

Figure 6-5: Receiver Locations for Rural Environment Testing (Source: DigitalGlobe and Google)

The terrain profile between the transmitter and test site 9 is shown in Figure 6-6.

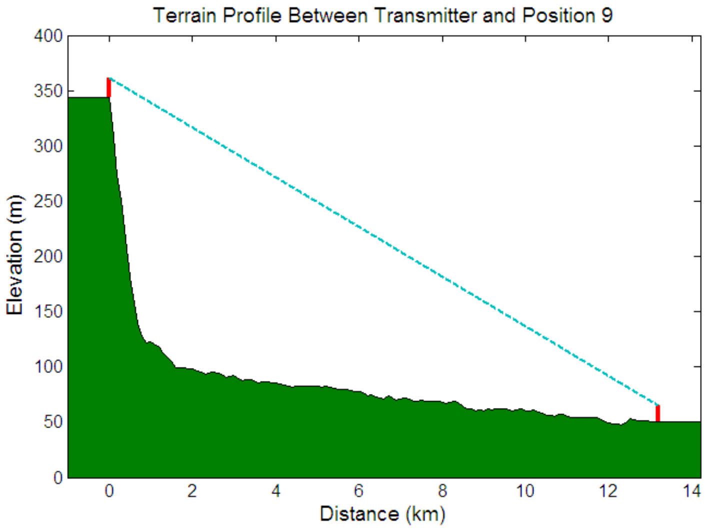


Figure 6-6: Terrain Profile between Transmitter and Position 9 (node height not to scale)

6.1.3.1 Predicted Results

The predictions given by the WSN model are shown in Table 6-4 with the closest fitting predictions shaded in grey. The WSN model gives the closest match for the case where there is a direct ray and a ground reflection with no multipath components. In most cases, the WSN model gives results within 3 % (1.7 dB) of the measured values, except in the cases for positions 2, 7 and 8. In these positions, it is predicted that the received signal is affected by a multipath component or a terrain feature that is not shown in the digital elevation data used for the WSN model.

Table 6-4: Comparison of Measured and Predicted Results for Rural Tests

Posn.	pR_{meas} (dBm)	Dist. (km)	Num. reflections	0	1	2	3	4	5	Plane-Earth model
1	-56.3	2.8	pR_{mean} (dBm)	-46.2	**-54.6**	-54.5	-54.3	-54.3	-54.0	-55.1
			SNR_{mean} (dB)	44.3	35.9	36.0	36.2	36.2	36.5	35.4
			σ (dB)	0.00	0.56	2.52	3.17	3.87	4.21	0.00
2	-80.3	4.0	pR_{mean} (dBm)	-50.6	**-61.0**	-60.9	-60.7	-60.5	-60.5	-60.6
			SNR_{mean} (dB)	39.9	29.5	29.6	29.8	30.0	30.0	29.9
			σ (dB)	0.00	0.59	2.78	3.36	4.04	4.78	0.00
3	-67.3	5.5	pR_{mean} (dBm)	-54.3	**-66.8**	-66.6	-66.2	-65.8	-65.8	-65.8
			SNR_{mean} (dB)	36.2	23.7	23.9	24.3	24.7	24.7	24.7
			σ (dB)	0.00	0.62	3.25	3.94	4.99	5.21	0.00
4	-70.4	6.7	pR_{mean} (dBm)	-56.5	**-70.4**	-70.2	-69.7	-69.1	-68.9	-69.1
			SNR_{mean} (dB)	33.9	20.1	20.3	20.8	21.4	21.6	21.4
			σ (dB)	0.00	0.63	3.19	4.37	4.63	5.08	0.00
5	-73.8	8	pR_{mean} (dBm)	-58.5	**-73.6**	-73.2	-72.8	-72.2	-71.7	-72
			SNR_{mean} (dB)	32.0	16.9	17.3	17.7	18.3	18.8	18.5
			σ (dB)	0.00	0.65	3.65	4.56	4.99	5.28	0.00
6	-77.2	10.1	pR_{mean} (dBm)	-60.9	**-77.6**	-77.0	-75.9	-75.0	-74.3	-76
			SNR_{mean} (dB)	29.6	12.8	13.4	14.6	15.5	16.2	14.5
			σ (dB)	0.00	0.66	4.18	4.98	5.49	5.58	0.00
7	-76.3	11.2	pR_{mean} (dBm)	-63.5	**-81.0**	-80.4	-79.6	-78.7	-77.5	-77.7
			SNR_{mean} (dB)	27.0	9.4	10.1	10.8	11.8	13.0	12.8
			σ (dB)	0.00	0.66	4.08	5.35	5.58	5.68	0.00
8	-77.5	12.2	pR_{mean} (dBm)	-64.7	**-82.8**	-81.8	-81.0	-80.2	-78.9	-79.2
			SNR_{mean} (dB)	25.8	7.72	8.68	9.47	10.3	11.6	11.3
			σ (dB)	0.00	0.68	4.12	5.51	5.74	5.98	0.00
9	-84.2	13.2	pR_{mean} (dBm)	-65.6	**-84.2**	-83.1	-82.6	-81.4	-80.3	-80.4
			SNR_{mean} (dB)	24.9	6.24	7.37	7.88	9.10	10.2	10.1
			σ (dB)	0.00	0.67	4.18	5.49	6.11	5.75	0.00

In six out of the nine measurements, the WSN model gave a closer match to measured data than the Plane-Earth model. However, note that the Plane-Earth model had better accuracy in the rural environment than it did in the suburban environment. This is due to the fact that there are less multipath components caused the absence of buildings in the vicinity of the link and the direct path being well clear of the earth for most of the link. A comparison between the WSN model, the Plane-Earth model and the measured data is shown in Figure 6-7.

It is shown in Table 6-4 that the received signal strength and signal-to-noise ratio are very low at the 13.2 km position. The signal-to-noise ratio is below the 7.59 dB threshold derived in section 6.1.2, above. Therefore, it is probable that bit errors will have occurred at this point. It is likely that during measurements, no bit errors occurred in the time that the packet was received and therefore the signal strength was able to be captured. This is also a possible cause of the variability noticed in Figure 6-7 at low signal levels.

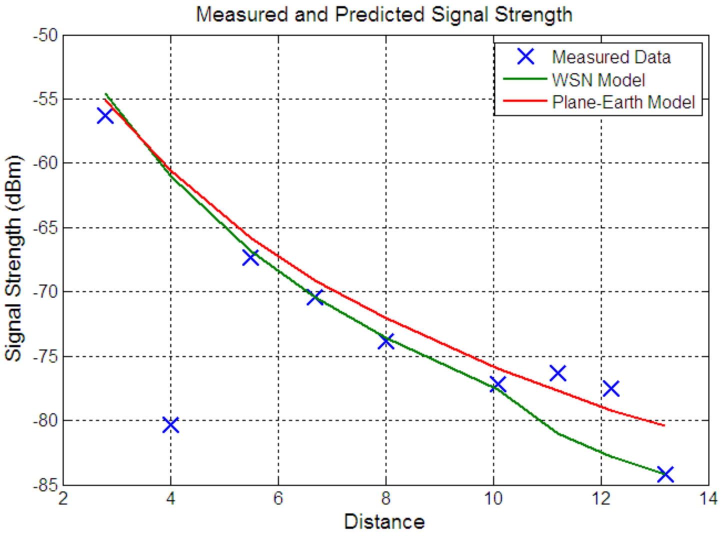


Figure 6-7: Comparison of Propagation Models and Measured Results

6.1.4 Ground-Level Testing

Ground-level testing was performed by installing the transmitter by the side of a rural road and measuring the signal strength with a car-mounted receiver node. The transmitter was installed on a 1.8 m star-picket with four radials, as shown in Figure 6-8. Note that the road has a slight rise and the receiver was placed beyond the rise.

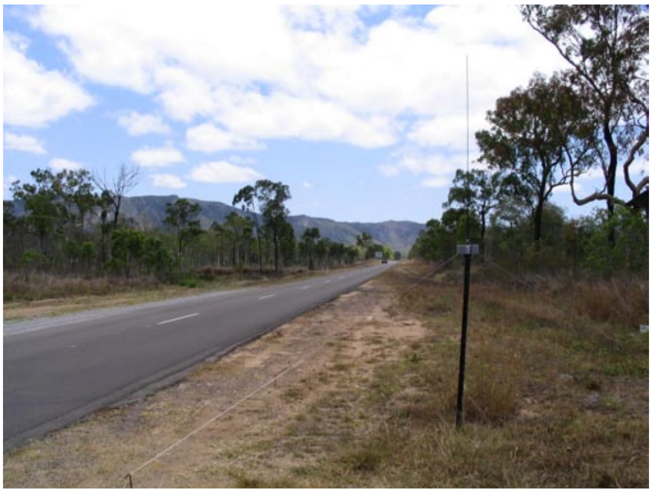


Figure 6-8: Transmitter for Ground-Level Tests

The results of the field tests are shown in Table 6-5, which demonstrates that a signal of -71.1 dBm was received at a distance of 850 m and no packets were received at 1.1 km.

TABLE 6-5: RESULTS OF GROUND-LEVEL TESTS

Test Position	Latitude	Longitude	Distance (km)	Elevation (m)	Signal Strength (dBm)
Transmitter	19°19'42.16"S	146°31'16.90"E	-	-	-
1	19°19'57.54"S	146°30'57.09"E	0.75	+1.8	-68.5
2	19°19'59.48"S	146°30'54.40"E	0.85	+3.1	-67.8
3	19°20'3.71"S	146°31'16.90"E	1.10	+3.6	-

The terrain profile between the transmitter and test position 3 is presented in Figure 6-9 and shows that the terrain has a slight incline. The terrain in Figure 6-9 appears to have sharp edges, because the elevation points have 100 m spacing. This is the accuracy required by the PTP model.

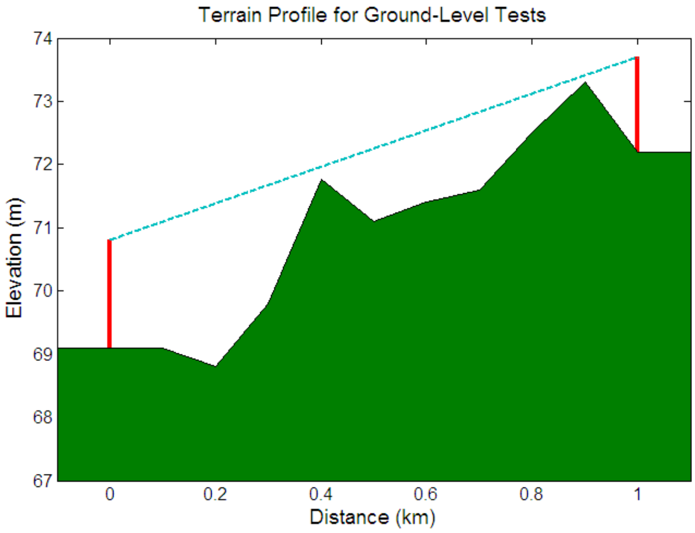


Figure 6-9: Terrain Profile for Ground-Level Tests

6.1.4.1 PREDICTED RESULTS

The predicted results, displayed in Table 6-6, show that the WSN model predicts the signal strength to be less than the measured value for the case of a direct ray and a ground reflection. When a multipath component is added, the WSN model predicts the signal strength to be too large. It is possible that the discrepancy is due to the PTP model predicting diffraction loss to be larger than it actually is. Figure 6-9 illustrates that the terrain profile for the PTP model has poor resolution for short links and may be a possible cause of error.

TABLE 6-6: COMPARISON OF MEASURED AND PREDICTED RESULTS FOR GROUND-LEVEL TESTS

Posn.	pR_{meas} (dBm)	Dist. (km)	Num. reflections	0	1	2	3	4	5	Plane-Earth model
1	-68.4	0.75	pR_{mean} (dBm)	-43.9	-77.4	-66.8	-64.0	-62.1	-60.6	-68.5
			SNR_{mean} (dB)	46.6	13.1	23.7	26.5	28.4	29.9	22.0
			σ (dB)	0.00	0.86	5.65	6.71	6.47	6.33	0.00
2	-71.1	0.85	pR_{mean} (dBm)	-44.8	-79.4	-68.2	-65.0	-63.0	-61.6	-67.8
			SNR_{mean} (dB)	45.6	11.1	22.3	25.5	27.5	28.9	22.7
			σ (dB)	0.00	0.86	5.43	6.42	6.37	6.46	0.00
3	-	1.1	pR_{mean} (dBm)	-46.8	-83.1	-69.3	-66.5	-64.9	-63.1	-71.4
			SNR_{mean} (dB)	43.7	7.38	21.2	24.0	25.6	27.3	19.1
			σ (dB)	0.00	0.86	5.13	6.50	6.03	5.98	0.00

The signal predictions by the Plane-Earth model were accurate for the first test position, but gave lower accuracy in the second position. Generally, in the ground-level tests, the Plane-Earth model gave predictions that were closer to the measured value than the WSN model. The results show that diffraction loss prediction for short paths with low antennas is difficult to predict. It is not recommended to use the WSN model for these cases.

Both models predicted that the signal-to-noise ratio would be above the required threshold to receive a packet at 1.1 km. However, no packets were received at this range. It is possible that a strong ground reflection was cancelling the direct ray at this point. Alternatively, the models may underestimate the diffraction loss when close to the Earth as in this test.

As shown above, the terrain at this location was not perfectly flat. It is expected that testing on perfectly flat terrain would show the range to be longer. This is explored further in section 6.3.1 using the WSN model and the plane-earth model.

6.2 WIRELESS SENSOR NETWORK TESTING

The Long-Range Wireless Sensor Network was tested by installing four nodes in a suburban environment. The nodes were configured to form an ad-hoc network and periodically send data to a base-station node, as would occur in a typical wireless sensor network application.

6.2.1 METHODOLOGY

Four JCUMotes were installed in the suburb of Annandale at the locations shown in Table 6-7. The base-station node was installed on the roof of the Electrical and Computer Engineering building at James Cook University (Figure 6-1) and all other nodes were attached to the existing television antenna poles on the private residences.

TABLE 6-7: NODE LOCATIONS FOR FIELD TESTING

Node Address	Latitude	Longitude	Distance from base-station (km)	Antenna Elevation (m)	Terrain Elevation (amsl)
0 (base-station)	19°19'53.59"S	146°45'29.50"E	0	13	43
3	19°18'58.17"S	146°46'1.72"E	1.9	4.8	19
4	19°18'49.44"S	146°46'54.56"E	3.1	3.6	13
1	19°18'31.10"S	146°46'43.40"E	3.3	4.0	13

Figure 6-10 shows the position of the nodes with the links indicated by yellow lines. The link between nodes 0 and 4 is dashed, because this link was occasionally operational (described below).

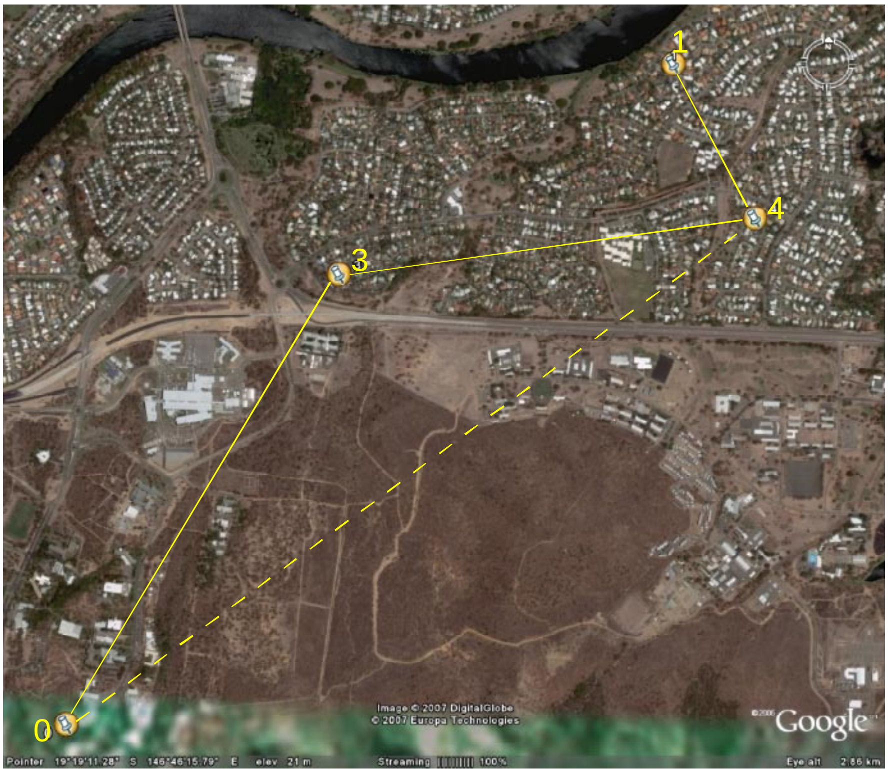

1
4
3
0
Image © 2007 DigitalGlobe
© 2007 Europa Technologies
©2006 Google
Pointer 19°19'11.28" S 146°46'15.79" E elev 21 m
Streaming 100%
Eye alt 2.86 km

Figure 6-10: Long-Range Wireless Sensor Network (Source: DigitalGlobe and Google)

The nodes were loaded with the `LRNet` (section 5.4.6) application to monitor the signal strength of the neighbouring nodes and send measurements to the base-station. Node 0 was connected to a PC which was running the MoteView software (section 5.4.6) to log all measurements to a database. The database was later queried to analyse the operations of the various links.

All nodes transmit +30 dBm of power, except node 1 which transmits at a level of +26 dBm. The nodes have between -78 and -81 dBm sensitivity, except node 1 which had -70 dBm sensitivity, as shown in section 4.10.2.

6.2.2 Received Signal Strength Measurements

A full four-node network was operational between the dates of 9th January 2007 – 18th February 2007, over which time 930 000 readings were logged in the database. The received signal strength of a neighbouring node was sampled each time a packet was received (every 8 seconds). The mean signal strength for each link was calculated for the period between the 12th January 2007 and the 28th January 2007. The measurements for each link are shown in Table 6-8.

Table 6-8: Long-Range Network Signal Strength Measurements

Tx	Rx	Rel. Elevation(m)	Dist. (km)	Tx Power (dBm)	Mean Signal Strength (dBm)	Std. Dev.	Num. Readings
0	3	32	1.9	+30.0	-67.3	1.71	20290
0	4	40	3.1	+30.0	-77.0	1.01	2981
1	0	-39	3.3	+25.5	-77.7	0.92	25
1	3	-6.8	1.4	+25.5	-73.5	0.92	5174
1	4	1.4	0.6	+25.5	-68.6	0.93	21009
3	0	-32	1.9	+30.0	-68.2	1.19	70570
3	4	8.2	1.5	+30.0	-67.7	0.73	20538
4	0	-40.4	3.1	+30.0	-75.0	0.86	15669
4	1	-1.4	0.6	+30.0	-62.9	1.18	33038
4	3	-8.2	1.5	+30.0	-66.2	0.94	31445

Each node transmits a measurement every eight seconds, except for node 0 which sends the measurements to the PC via the serial port. Node 0 only transmits routing updates which occur every twenty seconds (section 5.3.6). This explains the neighbouring nodes having a lower number of measurements of node 0's signal strength compared to other nodes. If a node receives packets from many neighbours in the eight second gap it will only report on one of the neighbours. This means that the results in Table 6-8 do not indicate the actual number of packets received from a particular node. The results only show the number of readings a node has sent to the base-station. An example of this is the link between node 1 and 4. Node 1 has reported more readings of node 4, than node 4 has of node 1. This is because node 1 has no other neighbours and therefore always reports the reception of packets from node 4. On the contrary, node 4 receives packets from three other nodes and therefore sends less reports of node 1.

Table 6-8 shows that some nodes would report receiving packets from a distant node on rare occasions. An example of this is that 25 packets were received from node 1 at the base station over a 3.3 km link. Node 1 has a lower power output than the other nodes and is separated from node 0 by the largest distance. Therefore, packets would have only been received when the radio propagation conditions were favourable. Node 1 has a lower sensitivity than the other nodes, which means that it only receives packets from the closest node, node 4.

The typical data measured on a clear day (dry ground) is shown in Figure 6-11, which presents the signal strength of packets received at node 0 from node 3. It is shown that there are no obvious diurnal variations in the received signal strength. The moving average has a peak-peak variation of approximately 3 dB. The figure shows there is larger variation after 7:00 pm. It is inferred that this is caused by the baby monitor interference (refer to 6.2.4).

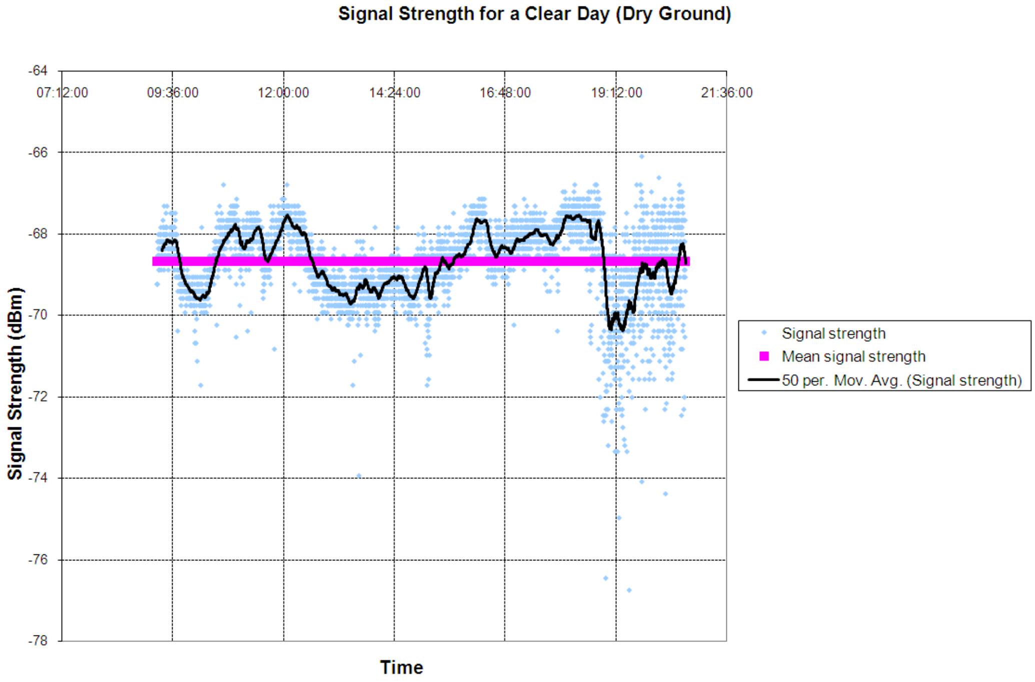


Figure 6-11: Measured Signal Strength on a Clear Day (Node 3 to 0)

6.2.2.1 PREDICTED RESULTS

The WSN model was used to predict the received signal strength for each link. The results are shown in Table 6-9, with the closest prediction shaded in grey. The worst of these predictions was within 2.5% (1.6 dB) of the measured value (4→1).

TABLE 6-9: COMPARISON OF MEASURED AND PREDICTED RESULTS

Link	pR$_{\text{meas}}$ (dBm)	Dist. (km)	Num. reflections	0	1	2	3	4	5	Plane-Earth model
0→3	-67.3	1.9	pR$_{\text{mean}}$ (dBm)	-47.4	-69.5	**-67.8**	-66.1	-64.6	-63.3	-54.7
			SNR$_{\text{mean}}$ (dB)	41.0	18.9	20.6	22.3	23.7	25.0	30.8
			σ (dB)	0.00	0.25	5.11	6.22	6.07	5.91	
0→4	-77.0	3.1	pR$_{\text{mean}}$ (dBm)	-53.2	-80.4	**-77.1**	-74.0	-71.8	-70.3	-64.4
			SNR$_{\text{mean}}$ (dB)	33.8	6.60	9.85	13.0	15.2	16.7	19.1
			σ (dB)	0.00	0.28	7.15	6.58	6.38	6.67	
1→0	-77.7	3.3	pR$_{\text{mean}}$ (dBm)	-57.2	-84.5	-80.8	**-77.5**	-75.2	-74.4	-69
			SNR$_{\text{mean}}$ (dB)	33.3	5.95	9.69	13.0	15.2	16.1	18.5
			σ (dB)	0.00	0.27	7.32	6.39	6.54	6.29	
1→3	-73.5	1.4	pR$_{\text{mean}}$ (dBm)	-52.8	-81.8	-75.7	**-72.0**	-69.4	-68.4	-66.1
			SNR$_{\text{mean}}$ (dB)	35.6	6.55	12.6	16.4	19.0	20.0	19.4
			σ (dB)	0.00	0.58	7.63	7.05	6.70	6.57	
1→4	-68.6	0.6	pR$_{\text{mean}}$ (dBm)	-40.2	**-68.8**	-64.2	-60.9	-59.0	-57.9	-61.0
			SNR$_{\text{mean}}$ (dB)	42.5	18.2	22.8	26.0	27.9	29.1	22.5
			σ (dB)	0.00	0.64	6.35	6.22	5.73	6.01	
3→0	-68.2	1.9	pR$_{\text{mean}}$ (dBm)	-47.4	-69.6	**-68.0**	-65.9	-64.4	-63.7	-54.7
			SNR$_{\text{mean}}$ (dB)	43.0	20.9	22.5	24.6	26.0	26.7	32.8
			σ (dB)	0.00	0.25	4.94	5.69	5.67	6.23	
3→4	-67.7	1.5	pR$_{\text{mean}}$ (dBm)	-47.8	-78.0	-71.2	**-67.8**	-65.7	-64.3	-63.7
			SNR$_{\text{mean}}$ (dB)	39.2	8.97	15.8	19.1	21.3	22.7	19.8
			σ (dB)	0.00	0.59	7.19	6.59	6.79	6.29	
4→0	-75	3.1	pR$_{\text{mean}}$ (dBm)	-54.2	-81.4	-78.4	**-75.0**	-73.6	-71.9	-64.4
			SNR$_{\text{mean}}$ (dB)	36.3	9.14	12.1	15.4	16.9	18.6	23.1
			σ (dB)	0.00	0.29	7.11	6.23	6.49	6.16	
4→1	-62.9	0.6	pR$_{\text{mean}}$ (dBm)	-40.2	**-64.5**	-59.8	-56.4	-54.4	-53.2	-61.0
			SNR$_{\text{mean}}$ (dB)	38.6	14.3	19.0	22.4	24.4	25.5	14.5
			σ (dB)	0.00	0.63	6.12	6.22	5.71	5.62	
4→3	-66.2	1.5	pR$_{\text{mean}}$ (dBm)	-48.4	-78.6	-72.3	-67.9	**-65.6**	-64.3	-63.7
			SNR$_{\text{mean}}$ (dB)	40.0	9.79	16.1	20.4	22.8	24.1	21.8
			σ (dB)	0.00	0.59	7.23	7.16	6.63	6.41	

By comparing Table 6-9 with the rural test predictions in Table 6-4, it is evident that in the suburban environment, the WSN model gives more accurate predictions when multipath components are included in the model. In the rural environment, the WSN model gave the most accurate predictions when there were no multipath components in the model. The WSN model gives the most accurate predictions if the number of multipath components is known. To aid in this prediction, an empirical model was developed that predicts the number of multipath components in a suburban environment. This model is shown in the following section (6.2.3).

Table 6-10 was generated by comparing the optimal number of reflections from Table 6-9 with the link distance and number of buildings in the Fresnel zone (counted using Google Earth [60]). The table shows that generally the links with the higher number of buildings were more accurately predicted by the WSN model using a high number of reflections. This seems to correspond with the theory that the houses act as reflectors and generate multipath signals.

TABLE 6-10: OPTIMAL NUMBER OF REFLECTIONS COMPARED WITH DISTANCE AND NUMBER OF BUILDINGS

Number of reflections	Link	Distance (km)	Num. buildings	Buildings/km
0	1→4	0.6	9	14
	4→1	0.6	9	14
1	0→3	1.9	9	4.6
	3→0	1.9	9	4.6
	0→4	3.1	11	3.5
2	4→0	3.1	11	3.5
	1→3	1.4	16	11
	3→4	1.5	23	15
	1→0	3.3	30	9.0
3	4→3	1.5	23	15

Table 6-9 also showed that the weaker links (such as 1→0, 1→3 and 4→0) are generally predicted most accurately using a higher number of multipath components. Table 6-8 demonstrated that there are less packets reported for the weaker links which means that the weaker links are only operational at certain times.

In order to successfully receive an LRNet packet of 33 bytes, a bit error rate less than 3.78×10^{-3} is assumed necessary. Using equation (24) from section 4.10.2, the required SNR was determined to be 8.53 dB. Table 6-9 shows that the weaker links are below or close to this threshold when the link is modelled with no multipath components, but above the threshold when one or more multipath components are present. This shows that the link relies on the correct multipath conditions in order to achieve communications, thus explaining why the link is only operational at certain times.

Table 6-9 also showed predictions generated by the Plane-Earth model. In all cases, this model predicted the signal strength to be higher than measurements. The Plane-Earth model is designed to model the simple case of having a direct ray and a strong ground reflection. It does not model multipath reflections and is therefore not as accurate in a suburban environment as it was in the rural environment described in section 6.1.3.

6.2.3 ESTIMATING THE NUMBER OF MULTIPATH COMPONENTS

In order to accurately model a link, it is necessary to determine the number of multipath components. An empirical model was developed based on the results presented in Table 6-10. The model uses three parameters to determine the number of multipath components: the number of buildings in the Fresnel zone, the height of the receiving node and the distance the closest building is to the receiver.

Table 6-10 shows that some links, such as between nodes 0 and 4 were estimated with a different number of multipath components in each direction. In the case of the 4→0 link, the receiver (node 0) is elevated above the mean building height. It is inferred that the multipath signals received by a higher node are stronger, because they are less affected by diffraction from the buildings. Therefore, if the receiver is elevated, the model has the highest accuracy when extra multipath components are specified.

Other links with nodes at similar heights, such as between nodes 3 and 4 were also predicted with a different number of multipath components in each direction. By analysing the path in Google Earth [60], it was noted that node 3 has a very close building in the Fresnel zone. The neighbouring house is approximately 10 m from the node. Alternatively, the closest house to node 4 is approximately 50 m away. Therefore, it is inferred that if a node is close to another building, then the probability of receiving extra multipath components is increased.

It was also noticed that in scenarios where the receiver was in an open space, the mean signal strength was predicted with a lower number of multipath components. This is evident for the link between nodes 1 and 4 and also in the suburban range testing presented in section 6.1.2. In the range testing, it was noted that tests 1 and 2 were more accurately predicted with 1 multipath component. However, test 3, which was more open, was more accurately predicted with no multipath components.

In the suburban range tests, the WSN model predictions showed less multipath components than similar links in the LRWSN field test. This is demonstrated in Table 6-11 (suburban range tests are shown as 0→Tn) with link 0→T2. This link is predicted with 1 multipath component and 16 buildings and link 0→3 also has 16 buildings, but is predicted to have 2 multipath components. It is inferred that the cause of this is the low height of the receiver. The suburban range test sites were much lower than the surrounding buildings and it is probable that less multipath components were received because of this.

TABLE 6-11: SUBURBAN LINKS SHOWING NUMBER OF BUILDINGS AND THE CLOSEST BUILDING

Num. Reflections	Link	Dist. (km)	Num. Buildings	Closest Building (m)	Closest Building (%)	Rx Height (m)
0	1→4	0.6	9	40	6.7	3.6
	4→1	0.6	9	80	13	4
	0→T3	2.7	14	40	1.7	1.5
1	0→3	1.9	9	40	2.1	4.8
	3→0	1.9	9	120	6.3	13
	0→4	3.1	11	60	1.9	3.6
	0→T1	2.2	14	40	1.7	1.5
	0→T2	2.5	16	40	1.6	1.5
2	4→0	3.1	11	120	3.9	13
	1→3	1.4	16	10	0.71	4.8
	3→4	1.5	23	50	3.3	3.6
	1→0	3.3	30	120	3.6	4
3	4→3	1.5	23	10	0.67	4.8

Table 6-11 was generated to develop an empirical model to calculate the optimal number of multipath components from the link specifications. The *Closest Building* columns specify the approximate distance of the closest building to the receiver in the Fresnel zone. The percentage column shows the ratio between the closest building distance and the path length.

The empirical model was developed by first applying a building correction multiplier to links where the receiver is close to a building or is open. Receivers that were open were classed as having building closeness greater than 5 %. The receiver is less likely to receive multipath components and therefore the number of buildings is decreased by a tested correction factor of 20 %. Similarly, receivers that have a building in close proximity were classed as having building closeness less than 2% and the number of buildings was increased by 20%.

Using the corrected number of buildings, a line of best fit was calculated for all links of normal height (links with a receiver dramatically above or below the average building height were ignored). The resulting line of best fit is shown in Figure 6-12 and is defined by equation (34).

$$m \approx {}^{b}\!/_{8} - 0.6 \tag{34}$$

where *m* is the number of multipath components and *b* is the number of buildings in the Fresnel zone.

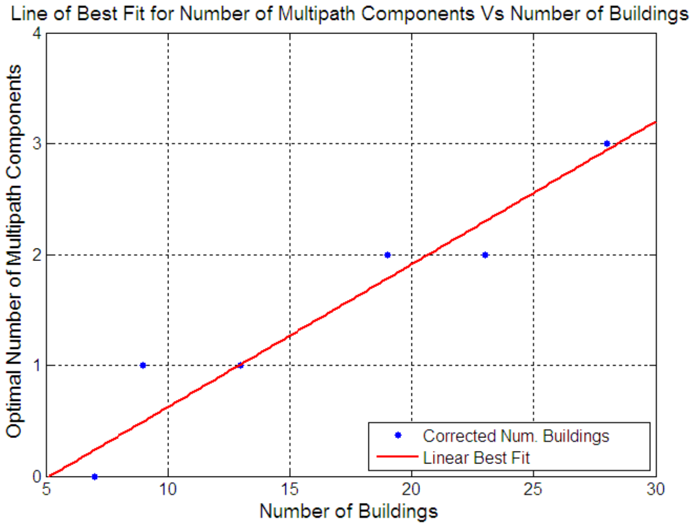


Figure 6-12: Line of Best Fit for Number of Multipath Components Vs Number of Buildings

The model compensates for elevated receivers by adding one extra multipath component if the node height is greater than twice the average building height. Compensation is also applied to low receivers by subtracting one from the number of multipath components if the node height is less than half the average building height. This model has been written as a MATLAB function and is named `EstimateNumMultipath.m`. The function was tested for the links in the sensor network and the range tests. The function correctly predicted the number of multipath components in all cases except for the link from node 1 to node 0. A possible cause of this is that the 1→0 link was only detected in a small number of cases (shown in Table 6-8) when the propagation conditions were ideal. The average propagation conditions could not be measured because the signal strength is usually below the node sensitivity level. Hence, the measurements only show the signal strength when it is above the mean.

It should be noted that this empirical model has been developed based on a small number of links. Further testing in other scenarios is required before this model can be considered reliable.

6.2.4 RADIO INTERFERENCE

During the field tests, it was discovered that baby monitors also operate in the 40.66 – 41 MHz ISM frequency band and can therefore interfere with the sensor nodes. Two different baby monitors made by Fisher-Price and Roger Armstrong were tested and both were found to operate at a selectable frequency of 40.67 or 40.69 MHz. The owner of the house where node 3 was installed said that the node transmissions were audible on the baby monitor. To alleviate this problem during testing, the node was wired to a power supply which the owner could switch off at night when the baby monitor was in use.

It was not expected that the node would interfere with the baby monitor, because it operates at a frequency of 40.85 MHz with a bandwidth of approximately 150 kHz (for a data rate of 9600 bps and a modulation index of two). The audible noise could be caused by the close proximity of the node and the large transmit power causing a small signal to occur in the channels of the baby monitor. It is also possible that the baby monitor uses a simple wideband receiver that has poor out-of-band rejection

The circuitry on the Roger Armstrong baby monitor (Figure 6-13) was inspected to investigate the receiver design and to also determine if any transceiver devices are used by manufacturers that may be useful in the JCUMote. The inspection showed that the baby monitor uses a simple design made from discrete through-hole components.

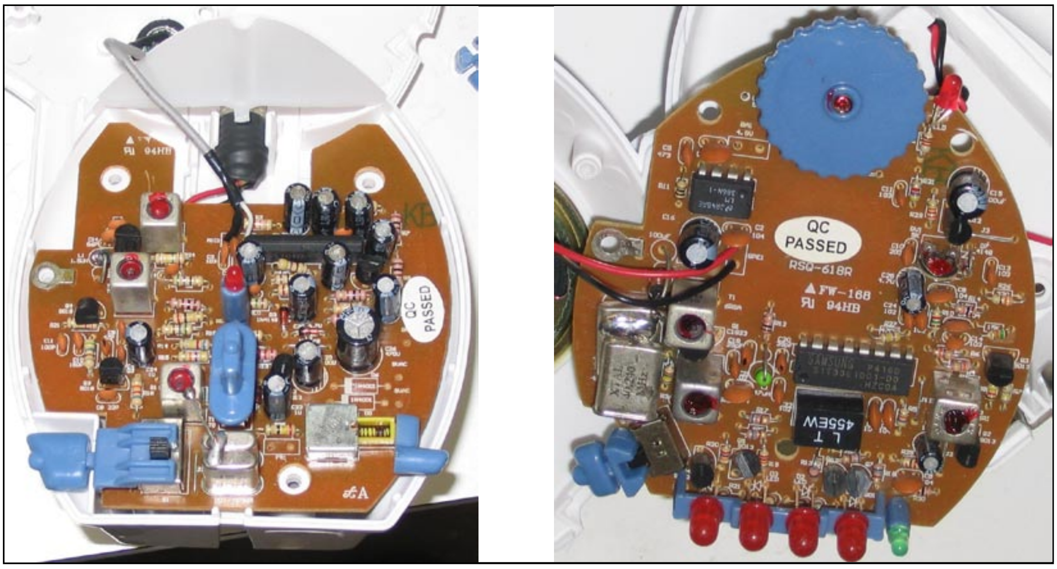


Figure 6-13: 40 MHz Baby Monitor (Left - Transmitter, Right - Receiver)

Field tests also showed that the baby monitor transmitter interfered with the wireless sensor network and caused the nodes to transmit fewer packets. It is expected that the decrease in packet transmissions was due to the collision avoidance mechanism which senses the received signal strength before transmitting. If the received signal strength is high, then the software assumes that another node is transmitting and waits a random time before attempting to transmit again.

The increase in noise also caused the adaptive squelch mechanism to raise the squelch level. Packets that were logged during the time that the node was on showed a sudden increase in the noise level. This is shown in Figure 6-14 which shows the noise level recorded one night when the node was left on. It is believed that the neighbouring household also had a baby monitor that generated the interference,

Figure 6-14 shows that the baby monitor was turned on at 19:24 and turned off at 06:47 the following day. In the hour prior to the baby monitor interface, 430 packets were received. After the activation of the baby monitor only 235 packets were received. It should also be noted that in Figure 6-14 a noise level of 0 indicates that no packets were received from neighbours during the time period between transmissions from node 3. The sudden increase of noise readings of 0 indicates that the baby monitor also reduces the effectiveness of the receiver.

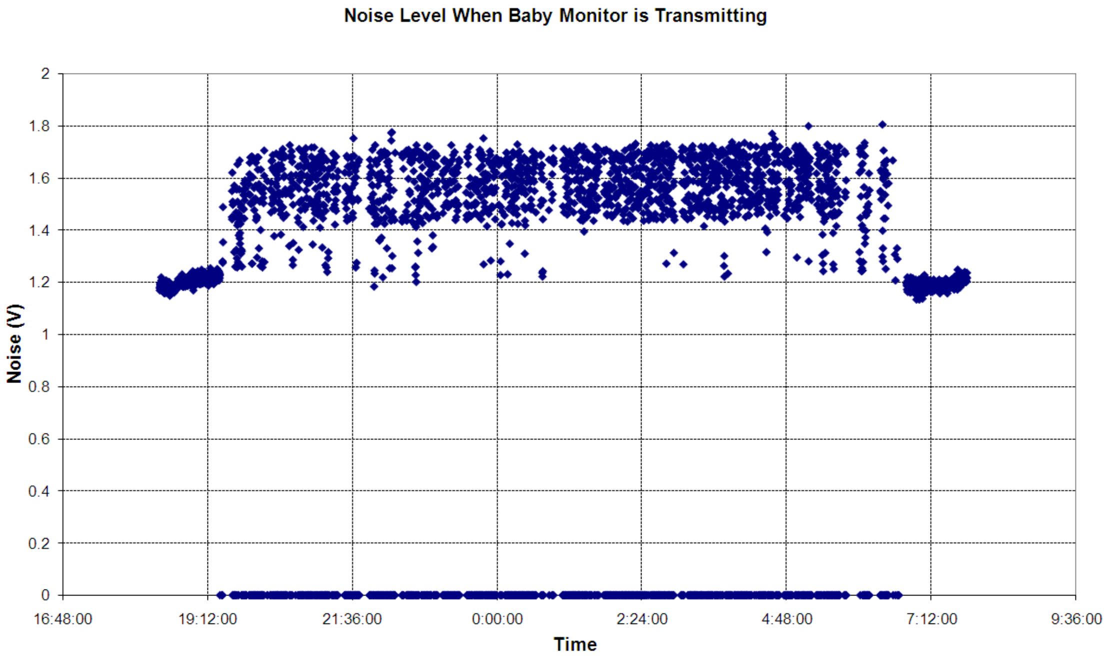


Figure 6-14: Noise Level When Baby Monitor is Transmitting

The amount of interference generated by the baby monitor could be reduced by using a narrow bandwidth receiver. The JCUMote uses a commercial IF filter with a 100 kHz bandwidth.

The baby monitor interference problem is not a major concern, because in a typical sensor network environment such as on an Australian cattle farm or the Great Barrier Reef, there are unlikely to be any baby monitors or similar interfering devices.

6.2.5 Routing Protocol Observations

The long-range wireless network was installed for 41 days and the network performance was logged in the database. A number of routing observations were made from the network database and are discussed in this section.

6.2.5.1 The 'Parent-Loss' Problem

Occasionally a node would receive unaddressed packets from a child. In this case the child node does not know the address of the parent and sends a broadcast message. This is illustrated by Figure 6-15, which shows the signal strength of packets received by node 0 which were sent by node 3 (a signal strength of 0 means that no packets were received from any neighbouring nodes). The blue dots represent all packets that were received by node 0 from node 3 (node 3 is usually the only node in range). These include addressed and broadcast messages. The red dots represent packets that were sent by node 3 and were correctly addressed to node 0 to be forwarded to the PC. These packets are logged in the database as readings from node 3. It is evident in Figure 6-15 that at times such as between 15:00-16:00, the base-station receives messages from node 3, but they have no parent address specified. Node 3 is usually responsible for forwarding packets from the other nodes. During times where node 3 'loses' its parent, the network packets are not forwarded to the base station, because they have no parent address.

The cause of the parent-loss problem is possibly due to node 3 not receiving some of the transmissions from node 0. The routing algorithm relies on the routing updates (sent every 20 seconds) to determine the link quality to the parent. If a node does not receive the route updates, then it will reduce the quality estimate for the link. If a sufficient number of route updates are missed then the node will drop the current parent and wait to find a new one.

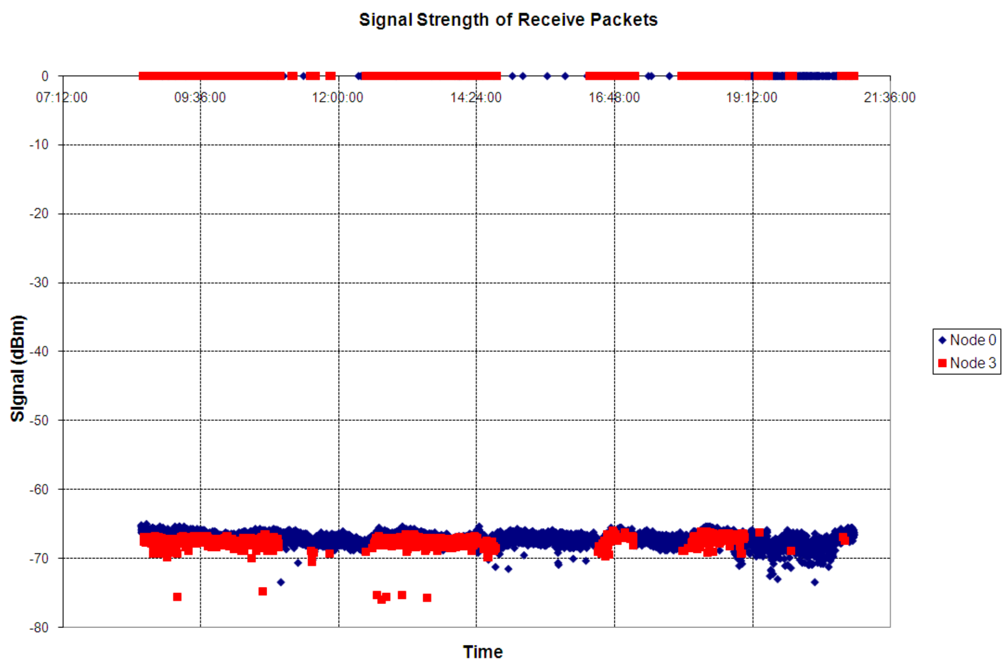


Figure 6-15: Received Signal Strength of Packets Received. Shows Node 3 Drops Out

This problem also occurs with node 4. Occasionally, packets are received by node 0 from node 4, but they are not addressed to node 0 and are therefore not forwarded to the PC.

Also note in Figure 6-15 that towards the end of the graph (after 19:00) node 0 starts reporting that no packets are received from the neighbouring nodes. This is caused by the baby monitor, which interferes with node 3 and reduces the number of transmissions due to the carrier detect mechanism.

The cause of 'parent loss' problem is not conclusive, because the time of the occurrence does not follow any pattern. Initially, it was thought that the transmitter on node 0 may have been faulty, but it was tested and no problems were identified. It is also possible that the radio propagation conditions may occasionally prevent the route update transmissions from node 0 reaching nodes 3 or 4. Interfering devices such as baby monitors may also reduce the number of route updates received by a node.

This problem may be alleviated by adding another node in range of nodes 0 and 4. When node 3 cannot determine a path to the sink, then the other node can be used to forward traffic.

Further investigation of the Mintroute routing protocol, or an alternative, is also required to determine if the parent selection mechanism is too strict. Existing research usually focuses on developing software to suit dense networks where there are multiple paths to the parent. In the long-range network, there is only one link available, so the protocol should choose any link that is available, regardless of the link quality.

It should also be noted that the MAC layer does not use Acks (acknowledgements). An acknowledgment scheme allows receivers to transmit a short ‘Ack’ to notify the transmitter of a successful reception. The incorporation of Acks with a retransmission scheme would allow the parent node to rebroadcast messages to ensure that child nodes receive the periodic route update messages.

6.2.5.2 ALTERNATIVE ROUTE SELECTION

The route selection mechanism of the routing protocol was observed to be functioning correctly, since on a number of occasions packets were routed directly from node 4 to node 0 without being forwarded through node 3. An example of this is shown in Figure 6-16, which was recorded on the 14th and 15th January 2007. This figure shows the parent of node 4 (green line) vs. time. Usually, node 4 would use node 3 as a parent (9am to 10pm). However, node 3 was turned off at night, so node 4 routed packets directly to node 0 (between midnight and 9am on the first day and approximately 7am on the second day). When node 3 was switched on in the morning, node 4 resumed forwarding packets through node 3 as it is a higher quality link. It should be noted that packets from node 1 are also forwarded when node 4 is connected to the base-station.

In most cases node 3 would be connected direct to the base station. However, node 3 was witnessed to occasionally use node 4 as a parent to forward messages to the base station. This occurred on the 8th January 2007 and is shown in Figure 6-17. For this to occur, the link estimation mechanism in node 3 would have determined that the link through node 4 had a higher quality than the direct link to the base-station. The reason for this is unknown, but it is possible that interference or a radio propagation anomaly prevented node 3 from receiving the route update messages from node 0. At the time, node 4 had a direct link to the base-station, so the parent selection mechanism in node 3 determined that packets should be forwarded through node 4.

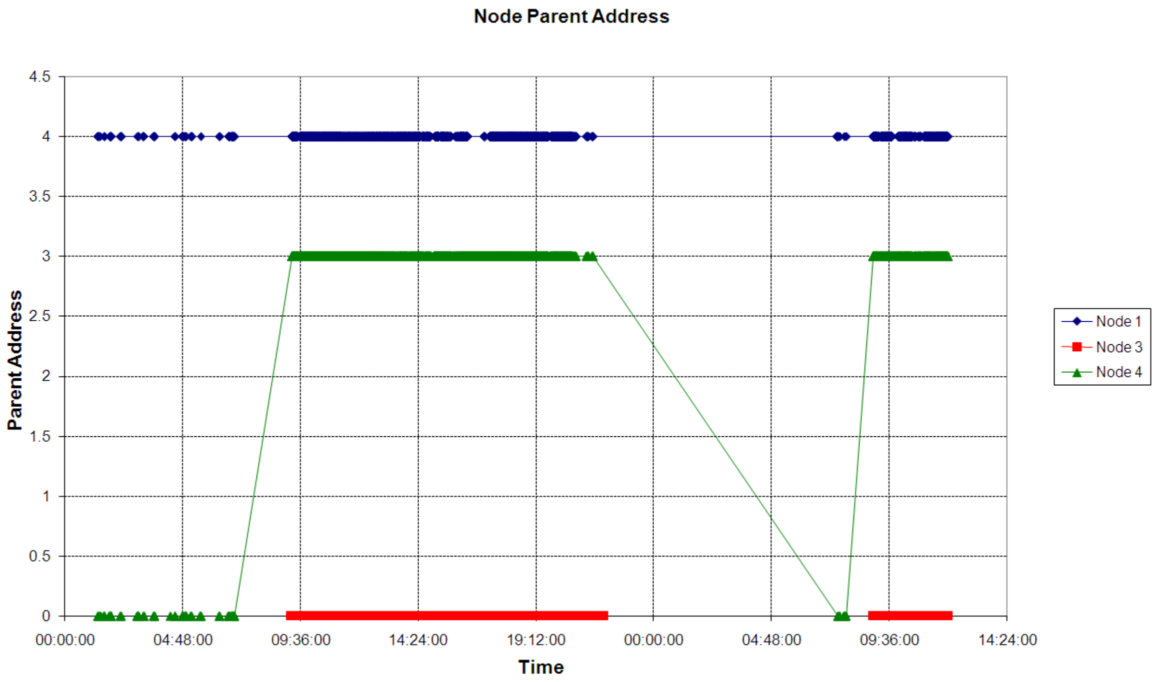


Figure 6-16: Node Parent Vs Time. Shows Node 4 Changing Parents

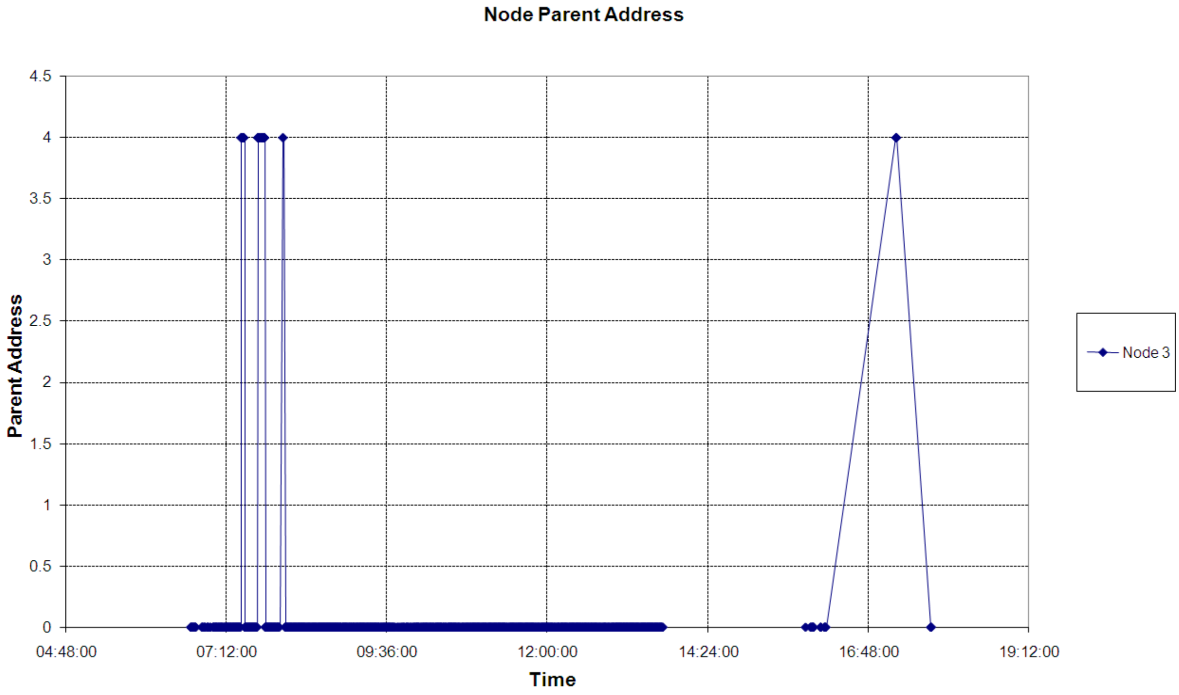


Figure 6-17: Node 3 Parent Vs Time - Shows Node 3 Changing Parents

6.2.6 NETWORK RELIABILITY MEASUREMENTS

When the network was operational, most of the packets were forwarded through node 3. Hence, the reliability of the network is basically dependent on the performance of this node. During the network field-test, it was noted that the main cause of long-term connectivity loss (greater than an hour) was due to the baby monitor interference or the node being turned off each night. The other main reason for losing connectivity was battery failure in nodes 1 or 4. When the battery failed in node 4, then node 1 would be disconnected from the network as node 4 was the path to the sink.

On one occasion, node 4 lost connectivity to the network due to the loss or corruption of the program code. The node was reprogrammed and regained connectivity. The cause of this problem is not known, but it highlights the need for an automatic reprogramming system such as Deluge (section 5.4.7). If Deluge was installed, the node would have reloaded a new program image from the external flash memory.

The other cause of connectivity loss is the parent-loss problem discussed above (section 6.2.5), where nodes would lose record of the current parent, because the link estimation mechanism determines that the link quality is low.

6.2.6.1 DAILY NETWORK RELIABILITY MEASUREMENTS

The network data from several typical days was used to produce the measurements shown in Table 6-12. These measurements were taken in the daytime period between the time that node 3 is turned on until the time that the baby monitor interference starts (on the 13/1/2007 node 3 was not switched on). The table shows the number of packets from each node that were logged to the database on each day. The packet loss is determined using the first and the last recorded sequence numbers from each day. The third column lists the number of drop-outs that were longer than ten minutes and the final column contains the length of the longest drop-out for each day.

In some cases such as the 14th and 15th January, there were no recorded drop-outs longer than ten minutes for nodes 3 and 4. Therefore, the small packet loss figure is due to the occasional loss of individual packets. This figure could be reduced by using acknowledgement messages and retransmissions. However, in the case of the prototype sensor network, this is not required, because the messages are sent every eight seconds which is very frequent. If messages were sent at a lower rate, then this problem would need attention, because it may mean that the user misses a crucial sensor measurement.

TABLE 6-12: DAILY NETWORK RELIABILITY MEASUREMENTS

Date	Node	Packets logged in database	Packet loss (%)	Drop-outs > 10 mins.	Max. Drop-out time
12/1/2007	3	4785	17.0	1	00:20:59
	4	4262	25.9	1	00:22:10
	1	4102	29.0	1	00:26:05
14/1/2007	3	4683	5.34	0	00:05:16
	4	4355	12.5	0	00:05:45
	1	3443	30.6	4	00:40:11
15/1/2007	3	4227	3.67	0	00:00:34
	4	3841	13.2	0	00:01:23
	1	3370	23.6	2	00:14:50
16/1/2007	3	3645	10.7	2	00:20:06
	4	3241	20.3	2	00:20:28
	1	2927	27.7	3	00:24:48

Table 6-12 also shows that the percentage of packets lost is related to the number of hops to the sink. This is expected, because if a message has to be transmitted several times, then there is a greater chance of an error occurring. Additional packet loss occurs when a parent node recovers from connectivity loss. The children nodes take a certain period of time to determine that a route to the sink is available. During this time, sensor readings are dropped.

It was determined that medium-term (greater than ten minutes) packet loss was caused by the parent-loss problem described above. In all cases, the parent node was able to receive transmissions from the child, but they were not always addressed correctly.

6.2.7 THE EFFECT OF RAINFALL

The sensor network was installed during the tropical wet season and was therefore subject to frequent rainfall. The rainfall had differing affects on the network performance and no conclusive observations could be made. Table 6-13 shows the network reliability measurements for node 3 with Townsville rainfall data provided by the Australian Bureau of Meteorology [87]. The rainfall data was collected at the Townsville Airport (19.2483^{o}S, 146.7661^{o}E), which is 9.2 km from the base station.

From these measurements, it can be observed that the rain caused network connectivity to be lost for long periods. During these periods it was noted that the loss in connectivity was due to the parent-loss problem discussed above. It is interesting to note that the packet loss was much larger on the 31st January than the 1st February, when much more rain was received. It was noted on the 1st February that the rain was constant with very heavy down-pours at 11:45 and 12:45. During these down-pours connectivity to node 3 was obtained, but during the lighter rain the parent-loss problem occurred. The cause of this is unknown, but it is possible that the heavy rain caused a change in the ground reflection or multipath reflections which enhanced the received signal. The effect of the rain on the signal strength is discussed below.

It was calculated that without the drop-out periods longer than ten minutes, the packet loss would have been 25.9%, 22.5% and 6.57% for the 31st January, 1st February and 4th February, respectively. This shows that the parent-loss problem is the main cause of packet loss.

TABLE 6-13: THE EFFECT OF RAINFALL ON NETWORK PERFORMANCE

Date	Rain (mm)	Packets logged in database	Packet loss (%)	Drop-outs > 10 mins.	Max. Drop-out time
14/1/2007	0	4683	5.34	0	00:05:16
15/1/2007	0	4227	3.67	0	00:00:34
16/1/2007	0	3645	10.7	2	00:20:06
31/1/2007	68.4	306	94.2	8	03:36:09
1/2/2007	197.8	2065	58.4	8	02:28:56
4/2/2007	29.4	2772	39.9	6	01:36:55

The effect of rainfall on the received signal strength was analysed by comparing the signal strength during rain with the mean signal strength on a clear day. On a clear day, the signal strength for the link from node 3 to node 0 was calculated to be -68.7 dBm (14th January). This was compared with the data collected on the 1st of February to give the plot shown in Figure 6-18. The mean received signal strength on the 1st of February was -67.6 dBm, which is marginally higher than the mean on the dry day. It is difficult to quantify the cause of this, but it is possible that the improvement in signal strength is due to the ground having a higher conductivity or the rain reducing the strength of the multipath reflections. It should be noted that the field trials were conducted during the tropical wet season and the ground moisture was not measured. It is likely that the soil was still moist on the 14th of January (but less moist than on the 1st February) from rain on the previous days. This soil moisture content is a possible cause of the small difference between the mean of the signal strength on the 14th January and 1st February.

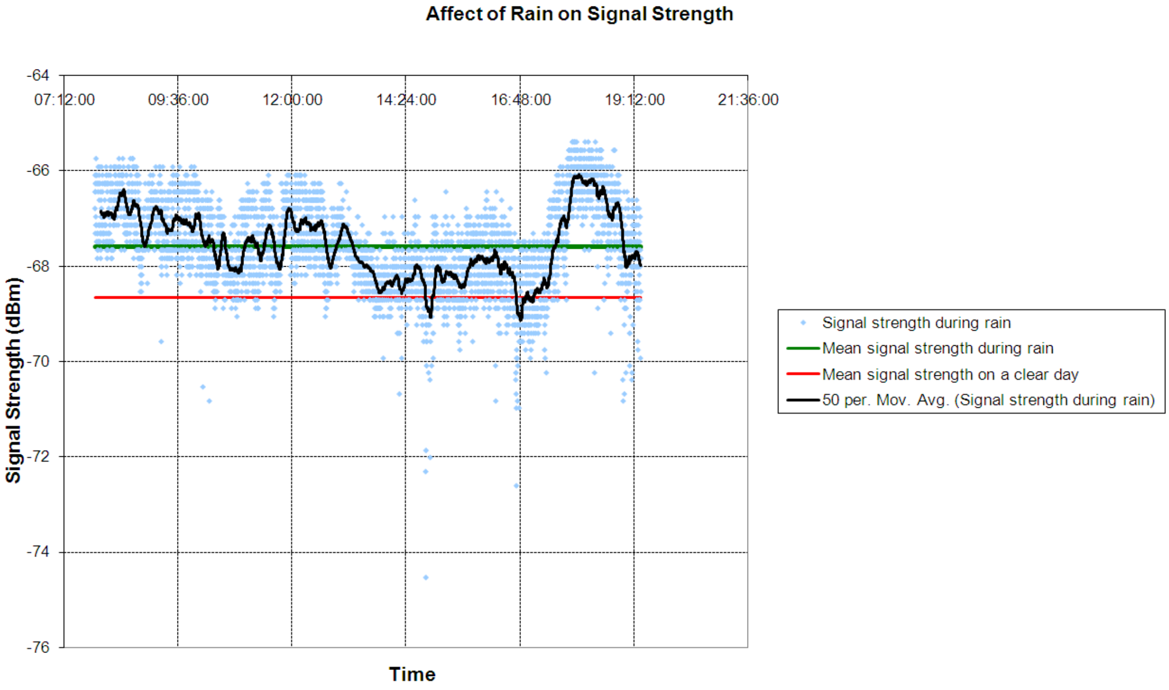


Figure 6-18: Effect of Rain on Signal Strength

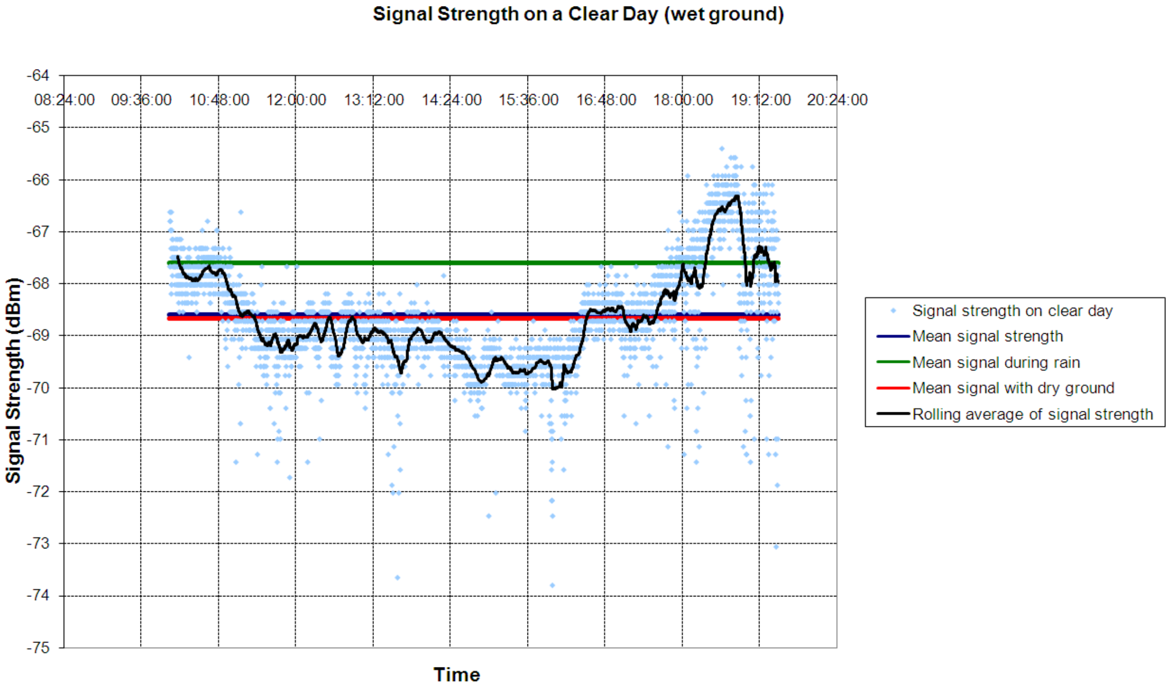


Figure 6-19: Signal Strength on a Clear Day with Wet Ground

The plot shown in was generated as a further investigation into the effect of the moist ground. The data in was collected on a clear day immediately following the days of rain (9th February 2007). On this day the ground was very wet and it is therefore likely to have had higher conductivity than on the dry day (14th January 2007) used as the mean in Figure 6-18. illustrates that the amount of moisture in the ground has little affect on the mean values of the received signal strength. Comparing the signal strength on the clear day with the mean from the wet day shows that the mean of the signal strength on a wet day is marginally higher. Comparing Figure 6-18 and shows the variability in the signal strength during rain is slightly larger.

Generally it appears that the rain has had a marginal affect on the mean signal strength. From this data it is difficult to determine if the rain was the cause of the parent-loss problem that is evident from the network reliability data presented in Table 6-13.

6.2.8 Solar Panel Operation

A solar panel was installed on node 4 as a proof-of-concept. The solar cell is a 6 V amorphous solar cell that is 152 × 152 mm in size and is quoted to produce 120 – 160 mA of current. The voltage produced by the solar panel on a clear day is shown in Figure 6-20. The data is unavailable at night, because the solar cell is connected to node 4 which relies on node 3 to forward the results to the base-station and node 3 is turned off at night because of the baby monitor problem (section 6.2.4).

The logs of battery voltage showed that the solar cell was not large enough to maintain the battery voltage. On the day shown in Figure 6-20, the battery voltage was 6.25 V at 19:30 and on the previous day the voltage was 6.28 V at this time. Therefore the battery lost 30 mV in 24 hours. It should be noted that this node is transmitting a large amount of data, because it sends a locally-originating packet every 8 seconds and also forwards the packets from node 1. Additionally, routing packets are transmitted every 20 seconds. Therefore, approximately 18 packets are sent every minute. In a typical sensor network application, it is unlikely that data packets will be sent at such a high rate. Therefore, the solar panel may be sufficient in an application where packets are sent less frequently. Alternatively, multiple solar cells could be added or a larger solar cell installed to maintain the battery charge.

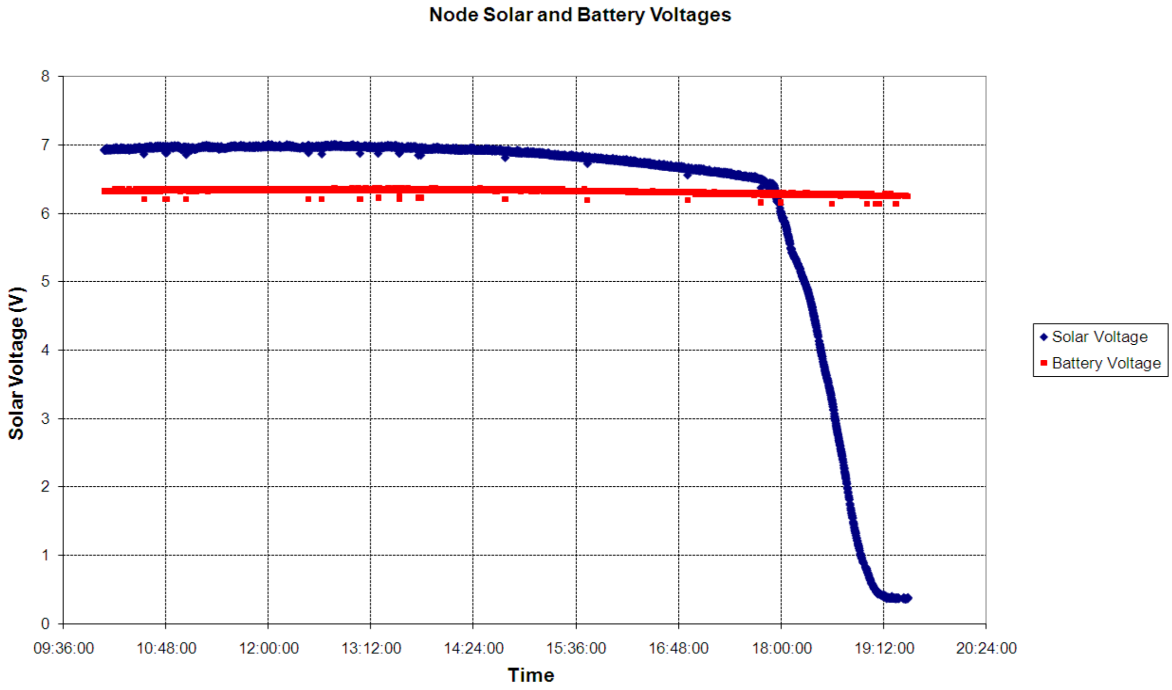


Figure 6-20: Solar and Battery Voltage on a Clear Day

It should be noted that nodes that are closer to the base-station transmit more data as they must forward packets from the network. This means that these nodes may require a larger solar cell capacity.

It was noted that the battery lasted 13 days before needing to be charged. During the 13 days, the area was subject to 6 days of rain or cloud so the solar cell was ineffective on these days. In the same period, the battery on node 1 (which has no solar panel) lasted 7 days. This node does not have to forward data from other nodes and transmits +25 dBm of power, compared to +30 dBm of power on node 4. Therefore, it would appear that the solar panel dramatically improved the life of the battery. It is difficult to quantify the improvement though, because the battery in node 1 was older than the battery in node 4.

6.2.9 Signal Strength Distribution

The signal strength measurements between nodes over the period of 12th January 2007 to 28th January 2007 were used to generate the probability density functions (PDF) shown in Figure 6-21 and the cumulative distribution functions (CDF) shown in Figure 6-23. The shape of the PDF curves appears to be similar to a Rician function with a large value of K (section 3.2.3). This occurs in situations where there is a large direct signal component and weaker multipath components. A subset of the time series data is presented in Figure 6-23 to Figure 6-25 which show measurements between the 22nd January 2007 and 26th January 2007.

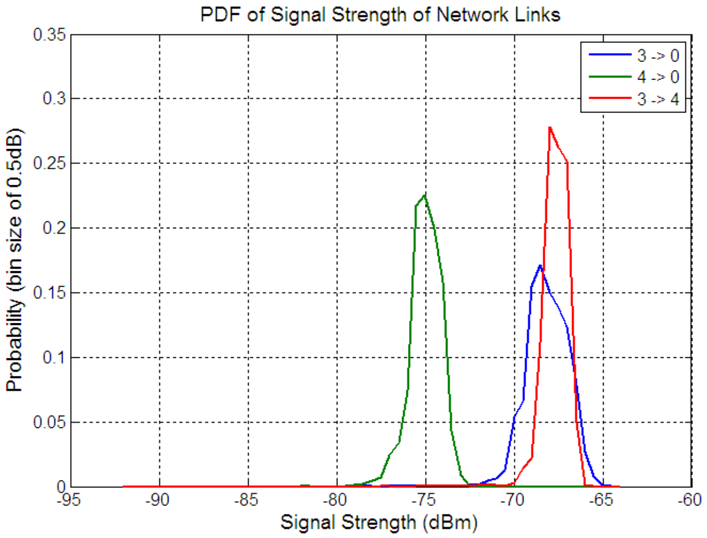


Figure 6-21: PDF of Signal Strength for Network Links

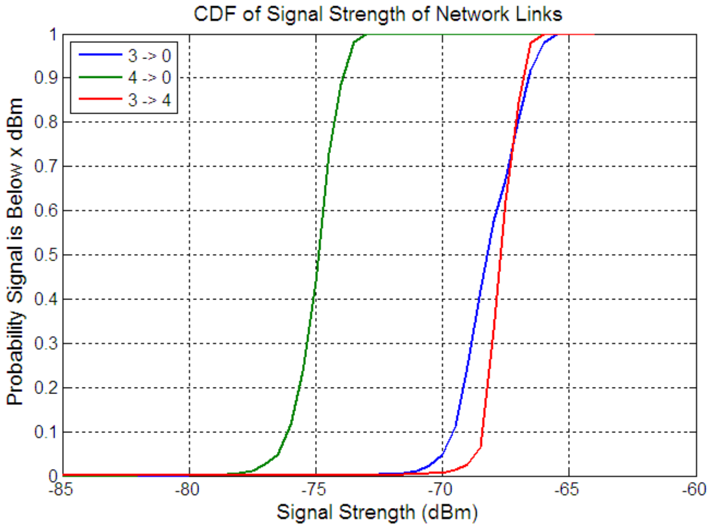


Figure 6-22: CDF of Signal Strength for Network Links

The mean and standard deviation of the signal strength received over each link is shown in Table 6-14. It is obvious that the link between nodes 3 and 4 has the smallest standard deviation. It is possible that this is because this link is the shortest (1.5 km). The link from node 4 to node 0 also has a low standard deviation. The signal strength for this link is close to the receiver sensitivity level and at low signal powers the packet may be ignored due to bit errors. Therefore, fading in the signal strength is undetected, hence giving a smaller standard deviation.

It is interesting that the link from node 3 to node 0 has the largest standard deviation, but is predicted to have only one multipath component. The large standard deviation is also evident in the time series data presented in Figure 6-23, which shows that signal fading occurred frequently on this link. This may be the cause of the parent-loss problem. Also, it was shown in Table 6-8 that this link has more measurements than the other links. Data was collected during periods (such as rainfall) when the parent-loss problem was occurring and data for other links (such as 3→4) was not available. The extra data over a broader time range may be another cause of the larger standard deviation value for this link, compared with the others.

TABLE 6-14: MEAN AND STANDARD DEVIATION OF SIGNAL STRENGTH

Link	Mean (dBm)	Standard Deviation (dB)
3 → 0	-68.2	1.19
4 → 0	-75.0	0.86
3 → 4	-67.7	0.73

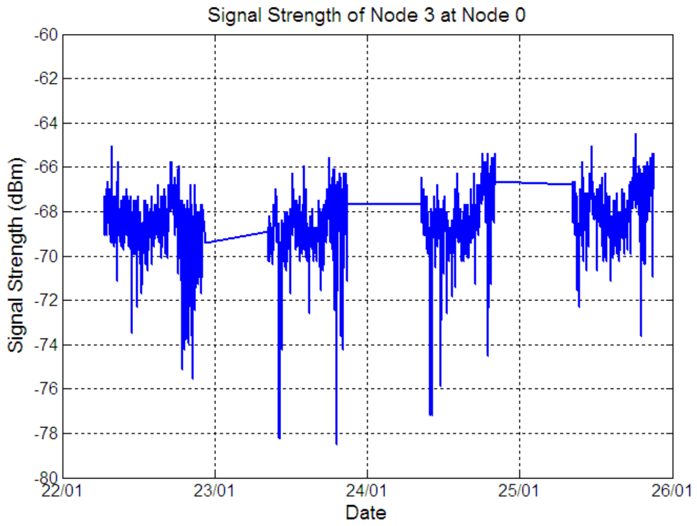


Figure 6-23: Signal Strength of Node 3 Recorded at Base Station (sample period approx. 16s)

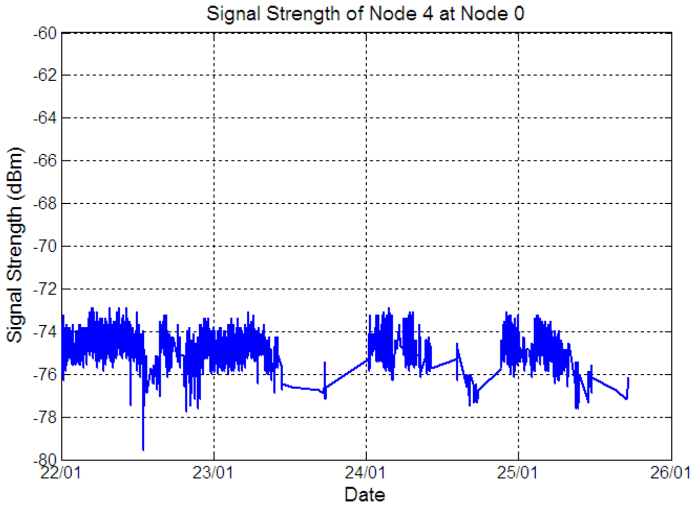


Figure 6-24: Signal Strength of Node 4 Recorded at Base Station (sample period approx. 16s)

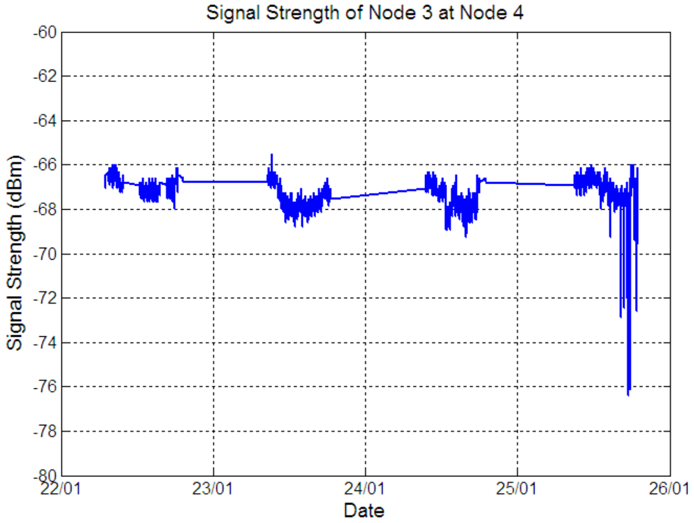


Figure 6-25: Signal Strength of Node 4 Recorded at Node 3 (sample period appox. 16s)

6.3 INCREASING THE TRANSMISSION RANGE

The transmission range of the nodes can be increased further by improving the receiver sensitivity. Section 4.10.2 showed that on average the JCUMotes have an approximate receiver sensitivity of -80 dBm. Some FSK transceiver ICs such as the Chipcon CC1000 [88] or the Nordic nRF903 [89] have a quoted sensitivity of -104 dBm (to achieve a bit error rate less than 10^{-3} at 9600 bps and 76.8 kbps, respectively). However, both of these ICs are not capable of 40 MHz operation, hence the selection of the Melexis TH7122 [62] on the JCUMote.

It was shown in Section 4.10.2.2 that the background noise level on the outskirts of a suburban area was measured to be -95.8 dBm over a 100 kHz bandwidth. If the background noise is -95.3 dBm, then the signal must be approximately -86.5 dBm in order to achieve an acceptable (8.78 dB) SNR to receive an LRNet packet. This section estimates the improvement in transmission range if a -87 dBm signal was able to be received.

6.3.1 Transmission Range Improvement at Ground Level

The transmission range at ground level was simulated using the WSN model and the plane-earth model. It was assumed that the terrain was flat (z = 0 at all positions) with no obstacles. The nodes were assumed to be elevated 1.8 m above the ground (effective antenna height is 1.8 m). The reflection coefficient was calculated using a random ground dielectric constant in the range of 2 to 7. The angle of incidence of the ground reflection is calculated using simple trigonometry and is in the range of 0.2^{o} to 0.06^{o} for 1 km to 3.5 km links, respectively. The multipath reflections are calculated by assuming there is an imaginary reflector on the flat terrain (discussed in 3.3.3).

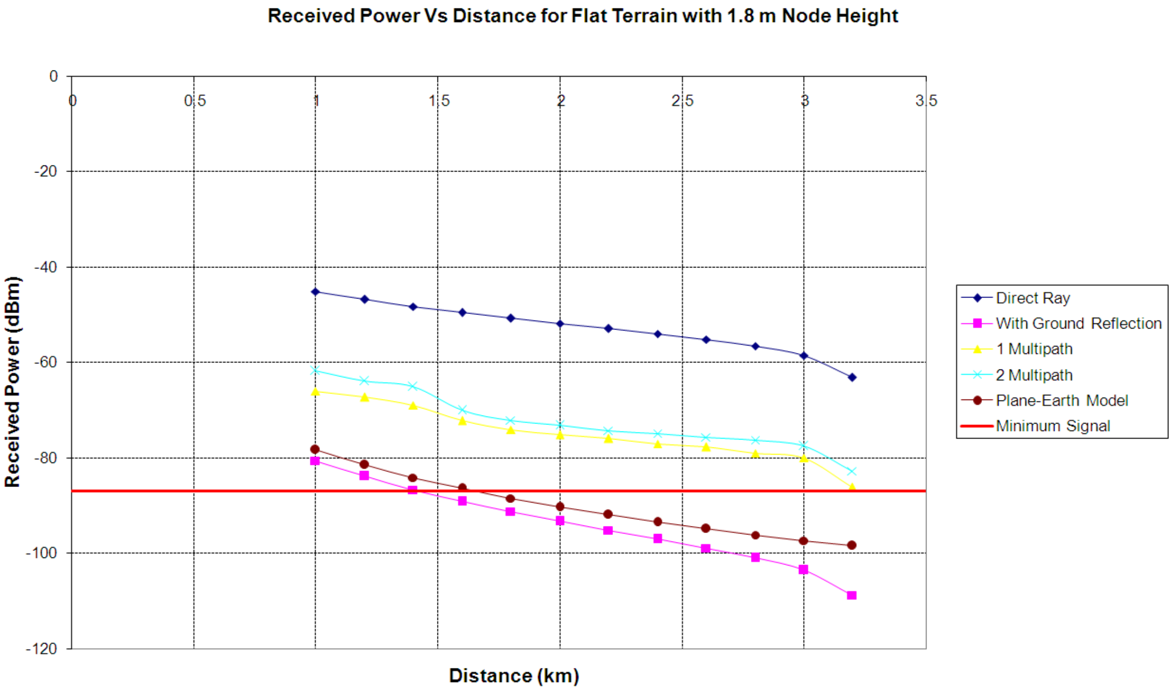


Figure 6-26: Simulation of Received Power Vs Distance for Flat Terrain with 1.8 m Node Height

The results of the simulation are shown in Figure 6-26. The received power falls below the receiver sensitivity level at approximately 1.4 km. This is assuming the worst-case scenario of a direct ray with a ground reflection and no multipath components. This is a likely scenario in a barren outback Australia environment. The current sensitivity of the JCUMote is approximately -80 dB, which gives approximately 1 km range according to Figure 6-26. This approximation seems reasonable, as it was shown in section 6.1.4 that a range of 800 m was measured over terrain that was not perfectly flat. The results from Figure 6-26 show that by improving the receiver sensitivity by approximately 7 dB, the range is increased by 400 m, which is a 40 % improvement.

6.3.2 Transmission Range Improvement in the Rural Environment

The rural environment discussed in section 6.1.3 was simulated to determine the possible increase in transmission range. Simulations were performed at 1km intervals with the receiver sites placed on the same road as used for testing (Figure 6-5). The elevation of the nodes was assumed to be 1.8 m. Figure 6-27 shows the results of the simulation with up to two multipath components presented.

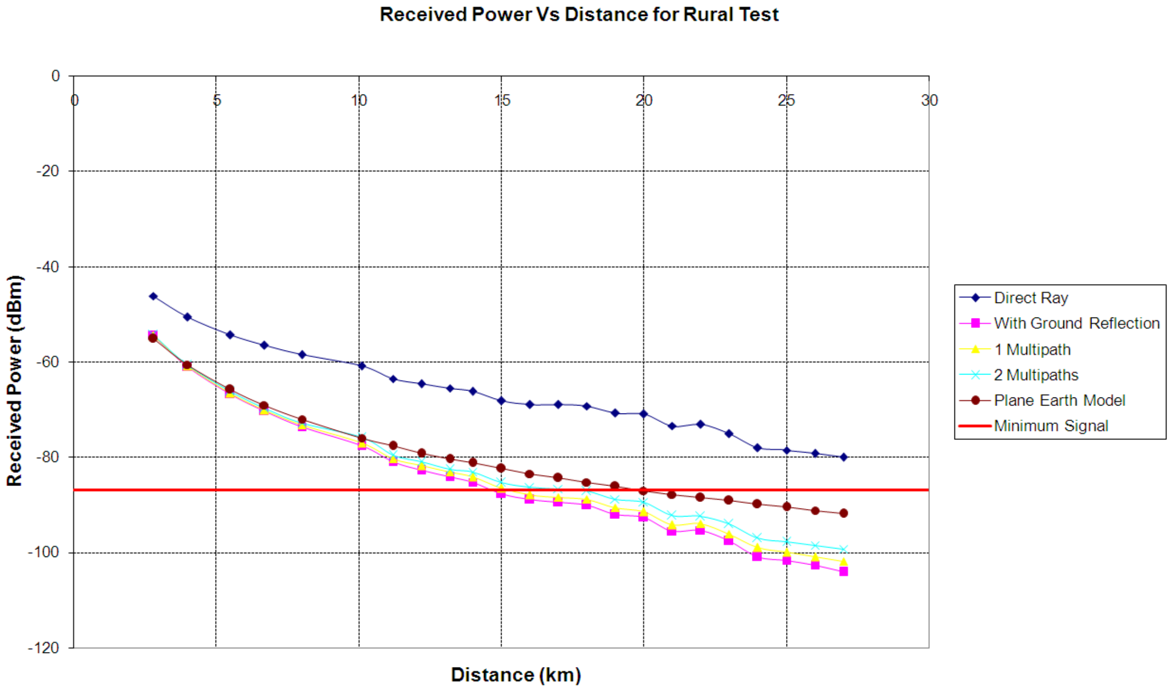


Figure 6-27: Simulation of Received Power Vs Distance for Rural Test with 1.8 m Node Height

The measurements in Table 6-4 showed that the simulator most closely matched the measured results for the case where there is a direct ray and a ground reflection with no multipath components. Figure 6-27 shows that in this case the range is increased from 13.2 km to 15 km (14 %) by improving the receiver sensitivity to -87 dBm. The case of a direct ray with a ground reflection and no multipath signals seems valid for the rural scenario where there are no large objects to act as reflectors. The plane-earth model seems to give more optimistic estimates that the WSN model, particularly at large distances. It was shown above in section 6.1.3 that the WSN model is more accurate in the rural scenario.

6.3.3 Transmission Range Improvement in the Suburban Environment

The transmission range improvement in the suburban environment was modelled by calculating the signal strength every 500 m on a bearing of 295^o from the position of node 3 in the wireless sensor network field test (section 6.2.1). The terrain on this bearing is fairly level. The nodes were assumed to be elevated 4 m, as is the typical case with a house-mounted node. Node 3 was assumed to be the transmitter. The predictions calculated by the WSN and plane-earth models are shown in Figure 6-28. In the worst-case direct ray with ground reflection, the range is increased from 1.5 km to 2.5 km (67 %) by improving the receiver sensitivity to -87 dBm.

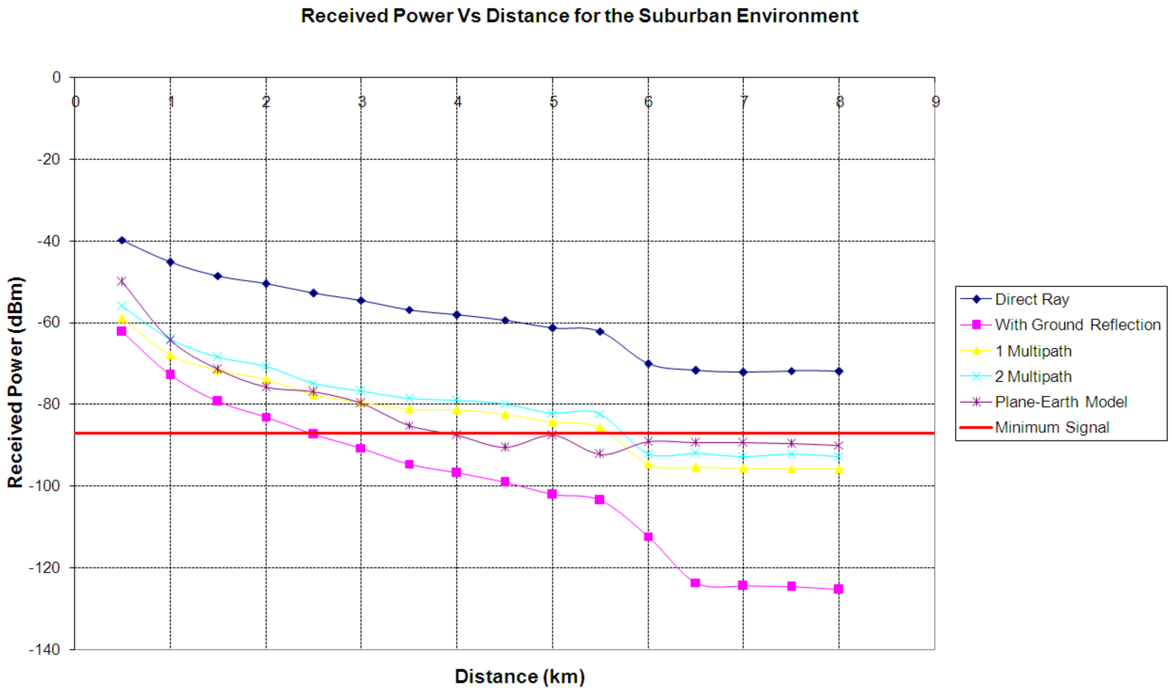


Figure 6-28: Simulation of Received Power Vs Distance for the Suburban Environment with Nodes Elevated 4 m

6.3.4 Investigation of Node Height

To estimate the effect of the node height, the received power was analysed over a fixed link (between nodes 3 and 4 in Figure 6-10) and the node heights were varied. The link is 1.5 km long and is fairly flat. The heights of transmitter and receiver nodes were set equal and varied as fractions of the wavelength (7.35 m). Figure 6-29 shows the resulting signal strength versus node height. It is evident that the node height has the largest affect on the received power when there is a direct ray and ground reflection with no multipath components. Figure 6-29 shows that increasing the node height from *λ/4* to *λ/2* can increase the received signal by 8 dB when a direct ray and ground-reflection are received only.

It is interesting to note that the model does not predict a large change in the received power when there is solely a direct ray or multipath components are present. This is because the node height mainly affects the ground reflection. When multipath components are added, the effect of the ground reflection on the overall received signal strength is reduced, hence the node height has less effect.

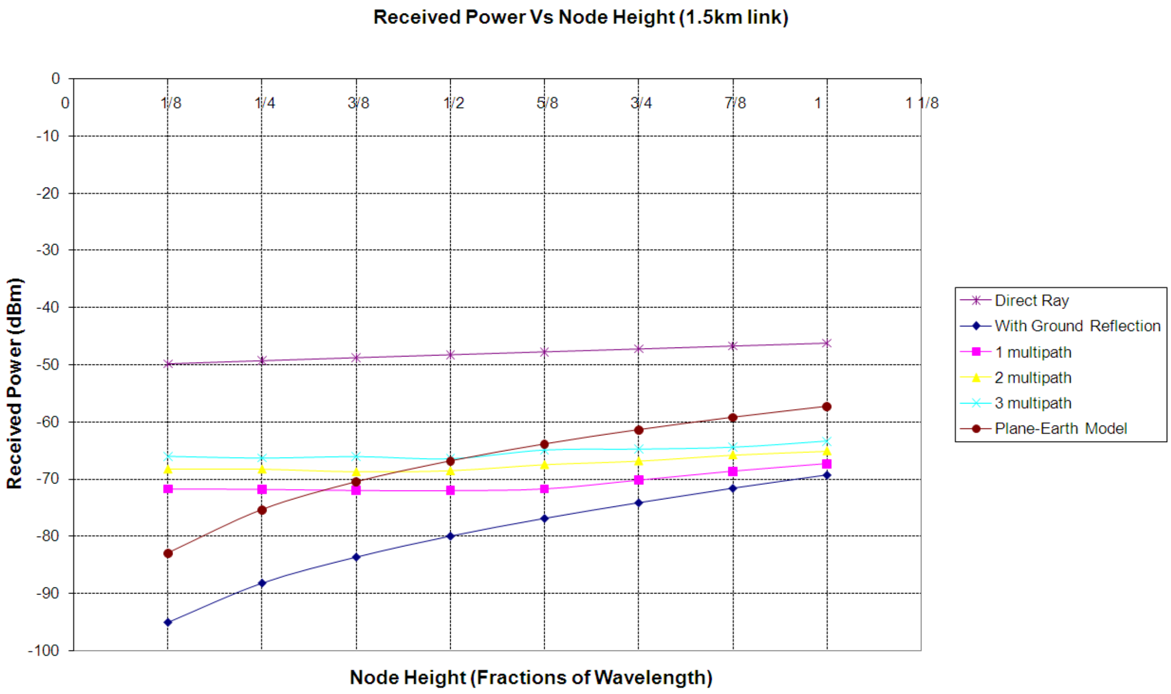


Figure 6-29: Simulation of Received Power Vs Antenna Height

If node height is increased, the signal strength increases due to three possible reasons. Firstly, the diffraction loss over nearby obstacles may be reduced. Secondly, if there is a large specular surface under the line-of-sight, then the received signal strength will be proportional to antenna height as shown in the plane-earth equation (3). Thirdly, it was predicted in section 6.2.3 that increasing the node height above the average building height increases the probability of receiving additional multipath components. This can increase the received signal strength considerably. As an example, a node at *λ/4* height that receives a direct ray and ground reflection only will receive a -88 dBm signal. This level is below the receiver sensitivity. If the height was increased to *λ/2* and a multipath component was received, the signal strength would be improved by 16 dB to -72 dBm. Hence, allowing packets to be received.

A summary of how the WSN model predicts the signal strength was provided in Table 3-2 of section 3.3. For the case of a direct ray and ground reflection, the signal strength is calculated by summing the direct and ground reflected vectors. This is equivalent to the two-ray model (3.2.2) with an additional diffraction loss component. It was stated in 3.1.2 that the two-ray model can be reduced to the plane-earth equation for the case of long links with low antenna heights. This explains why the WSN model with ground reflection has a similar curve to the plane-earth model. The difference of approximately 12 dB is caused by the diffraction loss.

6.3.5 Discussion

The predictions for increase in transmission range showed that an improvement up to 40 % is achievable (on flat terrain) by improving the receiver sensitivity by 7 dB. It was shown in section 6.3.4 that a similar improvement (8 dB) is possible by increasing the node height by a quarter-wavelength. Therefore, it can be estimated from Figure 6-26 (plot of received power vs. distance on flat terrain) that the combination of improving receiver sensitivity by 7 dB and raising the nodes by *λ/4* node height would increase the range from 1 km to approximately 2 km, which is a significant improvement.

6.5 Conclusion

This chapter has presented the results of signal strength measurements in the suburban and rural environments. It was shown that communication has been achieved over distances up to 13.2 km, which is considerably more than the 500 m range achieved using existing node technologies. This chapter also showed that a long range wireless sensor network has been deployed with the maximum operation link being 3.1 km long.

The results of field-testing were used to validate the novel radio propagation model presented in chapter 3. The results showed that in most cases, the WSN model gives predictions within 3 % (1.7 dB) of measured values in the rural environment. The WSN model also showed that in the rural environment there are not likely to be any multipath components present. However, in the suburban environment, the results showed that the accuracy of the model is dependent on the number of multipath components. A novel method was presented which allows the number of multipath components to be determined from the number of buildings in the Fresnel zone of a link, the closeness of buildings to the receiver and the height of the receiver.

The chapter also showed that significant improvements in transmission range can be achieved by improving the receiver sensitivity and raising the height of the nodes. It was shown from measurements that the background noise level at 40 MHz is fairly high. Hence, there is no point improving the node sensitivity by more than a maximum of 7 dB.

7 Discussions and Conclusions

The continuous advances in technology have allowed small, low-cost wireless sensor nodes to be developed. These nodes are being applied to an increasing number of monitoring applications. It was identified that a major limitation of these devices is their transmission range, which is usually several hundred metres. This restricts the devices to monitoring conditions over small geographical areas. The aim of this thesis was to investigate what changes are required to existing wireless sensor nodes to achieve long-range communication. This goal was achieved by developing a long-range wireless sensor node that is based on the Mica2 (section 2.4.1).

The first phase of this thesis proposed a radio propagation model that is suitable for long-range wireless sensor networks. This model showed the effects of multipath components and the surrounding terrain on the link performance. The model was refined using the results of the field tests and was shown to give predictions with 2 dB of measurements in most cases. It was shown that there are not likely to be multipath reflections in a rural environment, but these effects must be modelled in the suburban environment. A novel empirical model was developed that allowed the number of multipath components to be predicted from the number of buildings on a link, the closeness of buildings to the receiver and the height of the receiver.

The second phase of the thesis used the results from the radio propagation model to examine the feasibility of a long-range wireless sensor network and to propose the design of a long-range wireless sensor node. A node called the JCUMote was developed, which operates at 40.66 – 41 MHz and outputs 1 W of power. To achieve this, a novel radio transceiver was designed, which used an existing transceiver IC and a custom designed output power amplifier with receiver isolation network. The node development also involved the design and testing of several antennas. The final design consisted of a quarter-wavelength antenna surround by four radials.

The thesis also involved an investigation into the node software. A CSMA/CA medium access protocol and MintRoute routing protocol were implemented using TinyOS, a wireless sensor network operating system (section 5.2). Additionally, a Manchester encoder/decoder was implemented and tested at data rates up to 19.2 kbps. Additional software was tested such as over-the-air programming. A novel long-range WSN test application was developed which measured the signal strength of the neighbouring nodes and forwarded the readings to the base-station. The results of this were used to validate the radio propagation model.

The transmission range of the long-range wireless sensor nodes was tested in suburban and rural environments, where ranges of 2.7 km and 13.2 km were measured, respectively. Four nodes were tested as a long-range WSN, by installing them on the roofs of houses in a suburban environment. The network was operational for six weeks and over 900 000 readings were logged in the database. A maximum operation link of 3.1 km was observed.

During the testing, the routing protocol was observed to be operating correctly as it changed routes through the network depending on the radio propagation conditions. One problem was shown where nodes would stop addressing data to the parent node. It was predicted that the cause of this was the routing protocol, which would discard a parent if the link-quality was low. The implemented routing protocol was designed for a dense network, so it may need optimisation for long-range links. The testing also showed that commercial baby monitors operate at the same frequency as the long-range WSN. However, this is not a major concern, because there are not likely to be any baby monitors in the remote areas where a long-range WSN would typically be installed.

Overall, the field tests proved that existing sensor nodes can be adapted to achieve long-range communications.

7.1 Recommendations

This project set out to answer several research questions, but in the course of the investigation the developer identified further areas of possible future investigation. It was predicted in section 6.3.3 that the transmission range of the nodes could be increased by up to 67% (suburban environment) by improving the receiver sensitivity by 7 dB. This could be achieved easily by adding an LNA at the input to the receiver. The range could also be improved further by using an error correction scheme.

This thesis presented a proof-of-concept long-range WSN, which performed reliably. It is possible that this long-range WSN could be developed into a commercial system for long-range environmental monitoring. To do so, it was identified that the transceiver design would need to be refined so that less tuneable components are required. It would also be desirable to develop a more practical antenna that has similar performance to the tested antenna, but is easier to install.

Additionally, this project has shown that long-range communications can be achieved using a basic FSK modulation scheme and a CSMA/CA medium access protocol. This leaves scope for future investigation into the usage of more complex modulation schemes and medium access protocols on a long-range wireless sensor network.

7.2 Conclusions

This thesis has presented two main contributions to research: a novel long-range WSN radio propagation model and a new long-range wireless sensor node. It is possible that the former could be used in other RF engineering fields to predict the performance of radio systems, possibly at different frequencies. This model could also be integrated with existing network simulators to aid in the development of long-range WSNs.

This thesis has also presented a long-range wireless sensor node with considerably larger range than existing devices. This achievement dramatically increases the range of applications for WSNs as it allows nodes to be deployed over large areas. This will allow WSNs to be used for environment monitoring, which is particularly important as it may improve farming practices and also allow the monitoring of sensitive environments such as the Great Barrier Reef. It is possible that in the future long-range WSNs may be commonly used in applications that have not yet been imagined.