TECHNICAL UNIVERSITY OF CRETE
ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT
TELECOMMUNICATIONS DIVISION

Extended Range Scatter Radio Links with Embedded Radio

by

Georgios Vougioukas

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DIPLOMA OF

ELECTRICAL AND COMPUTER ENGINEERING

July 2016

THESIS COMMITTEE

Associate Professor Aggelos Bletsas, *Thesis Supervisor*
Associate Professor George N. Karystinos
Associate Professor Eftichios Koutroulis

Abstract

This work studies whether increased ranges of bistatic scatter radio communication are possible, especially when low-cost (in the order of 4 Euro), embedded receivers, originally designed for conventional radio (and not for scatter radio) are employed. It is found that for 13 dBm emitter transmission power, 246 meters scatter radio tag-to-reader distance is possible, with packet error rate (PER) less than 1%, while 268 meters are possible at the expense of increased PER, in the order of 10%. A prototype node employing the communication scheme suggested and powered by a supercapacitor along with a solar panel is also built. Additionally an experimental wireless communication link, utilizing the harmonics of a microcontroller's clock signal as the carrier is presented.
Thesis Supervisor: Associate Professor Aggelos Bletsas

Acknowledgements

First of all I would like to thank my supervisor Prof. Aggelos Bletsas for his guidance and support throughout this work and beyond that.

My friends and colleagues Spiros Daskalakis and Konstantinos Tountas for all the help they provided during my work on the laboratory. I would like to thank Konstantinos for the rides he gave me from Chania to Kounoupidiana (and the opposite), I owe you a beer.

My family, my close friends and especially, Spiros S., Nikos T., Thanasis T., Sotiris L. and Vaggelis T., nothing would be the same without them.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Internet of Things and Wireless Sensor Networks

Wireless sensor networks (WSN) have recently attracted immense research interest with a broad spectrum of possible applications. With the advent of the so-called Internet of Things (IoT) this interest has skyrocketed as by nature WSNs and IoT are closely related. It is estimated [1] that by year 2020, over 50 billion devices will be connected. To achieve a network of such scale, each device must communicate with the outside world using low cost and ideally ultra low power technology.

At the time being, the de-facto approach for making such a device capable of communicating, is utilizing commercially available modules that implement Bluetooth/WiFi or other proprietary radio standard. These modules due to facilitating active, radio frequency (RF) sub systems such as oscillators, mixers and power amplifiers, are expensive and have high power demand.

Using the aforementioned modules as the means for communication applies constraints to network scalability due to increasing the cost per device. In addition, high power consumption imposes the need for integrating a power source like battery on the device, which except further increasing both the purchase and maintenance cost, introduces the problem of disposing the battery at the end of its life, which if not done in a correct manner has environmental impact.

1.2 Scatter Radio

Instead of having modules like the aforementioned on each device (may referred as tag or node from now on) of the network, which radiate power to transmit information, a single active transmitter, called carrier emitter, may be employed to illuminate the tag/s. Each tag reflects the induced continuous wave (CW) signal coming from the centralized transmitter and modulates its information on that reflection. The tag employs an antenna connected to different termination loads via a RF switch or multiplexer. By changing the termination load, the antenna transmits the induced CW, coming from the carrier emitter, with altered phase or amplitude or both. That transmitted/reflected signal can be received by a dedicated reader and after processing, each tag's information can be recovered.

The scatter radio principle dates back to the 40s [2], but only recently has been utilized to specific applications. The most publicly known application of scatter radio is radio frequency identification (RFID) systems. These systems are used as means for identifying products, peoples, animals and performing payments. Their main drawback is the limited communication range which is in the order of 10 meters and this is due to the fact that the carrier emitter is collocated with the reader (Monostatic Architecture) and the RFID tag is powered by harvesting RF energy from the carrier emitter. However, by using a Bistatic Architecture (dislocated carrier emitter and reader),in order to minimize the interference from the emitter to the reader, scatter radio technology can be applied to the nodes of a WSN or an IoT device where long communication ranges are needed. To achieve communication using scatter radio the only thing needed, as mentioned earlier, on a node/device is a RF transistor/switch connected to an antenna. This minimises the power consumption, thus making it possible for the node to be powered by ambient sources using solar panels, thermoelectric generators (TEGs), high efficiency RF harvesters or all of them combined and thus eliminating the need for a battery. A bistatic setup can be seen in figure 1.1.

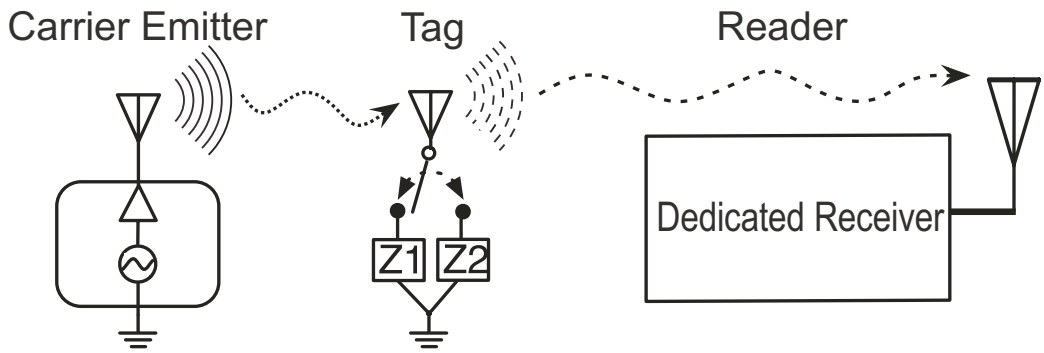


Figure 1.1: Basic bistatic scatter radio setup.

In this work conventional, low cost embedded receivers are used as readers for scatter radio signals instead of the more traditional Software Defined Radios (SDR). In Chap. 2 bistatic scatter radio principle is explained and a detailed description, along with experimental results, of the communication link implemented in order to achieve scatter radio communication with an embedded receiver, is given. In Chap. 3 the construction of an ambienlty powered scatter radio node, which implements the communication scheme described in Chap. 2, is presented. An experimental, short range communication link utilizing just a microcontroller unit and an embedded receiver is shown in Chap. 4. Work is concluded in Chap. 5 in which future plans for extending this work are presented.

Chapter 2

Long Range Scatter Radio Communication Link

2.1 Introduction

Recent advances in backscatter communication have enabled the development of ultra-low-cost and low-power wireless sensor/tags. Examples include large-scale environmental monitoring of air humidity [3] and soil moisture [4], which are both critical in agriculture, as well as short-range telemetry applications [5].

One key architecture for scalable backscatter communication is based on bistatic principles; the tag/sensor is illuminated by a carrier emitter and modulates information on its reflection coefficient; the signal is reflected by the sensor's antenna and received back by another unit, the reader. Thus, in bistatic architecture there are in principle three different entities involved, the carrier emitter, the tag/sensor and the reader, offering flexible setups. Bistatic scatter radio communication was first offered in [6–8] focusing on either on-off keying (OOK) or frequency shift keying (FSK) for scatter radio signals, highlighting the fundamental differences compared to conventional (Marconi) radio and also addressing issues relevant to the bistatic setup (e.g., carrier frequency offset between emitter and receiver). Ranges in the order of 140 meter were achieved, using commodity software-defined radios, while work in [9] further increased range by 10 more meters with careful coherent processing and small-block length channel codes. Work in [10] offered an example of bistatic scatter radio FSK, achieved using laboratory equipment and received by the bluetooth module of a smartphone.

This work studies whether increased ranges of scatter radio communi-

cation are possible, especially when low-cost, embedded receivers, originally designed for conventional radio (and not for scatter radio) are employed.

2.2 Link Implementation

2.2.1 Bistatic Scatter Radio

In order to achieve communication with an embedded receiver using scatter radio, we implement Binary-Frequency Shift Keying (B-FSK) modulation using just a switch connected to an antenna and a carrier emitter that illuminates the antenna with a CW signal. B-FSK modulation was chosen due to its high immunity to noise.

As briefly described in Section 1.2 and in Subsection 2.1, if a continuous wave carrier impinges an antenna connected to a RF switch/transistor which alters the antenna's termination loads, the antenna reflects the induced signal.

If the antenna terminals are connected to a load of value of Z_i then the reflection coefficient is,

$$\Gamma_{\mathrm{i}} = \frac{Z_{\mathrm{i}} - Z_{\mathrm{a}}^*}{Z_{\mathrm{i}} + Z_{\mathrm{a}}} \tag{2.1}$$

where Z_a is the antenna input impedance. When the switch selects a termination load between two distinct values, for instance Z_0, Z_1, due to 2.1, two reflection coefficients are achieved, Γ_0 when the antenna is connected to termination load Z_0 and Γ_1 when connected to Z_1. In this work these loads are,

- Z_0: Full short, i.e the antenna terminals are (ideally) shorted together.
- Z_1: Open circuit, i.e the antenna terminals are (ideally) floating.

The antenna is illuminated by a CW signal from the carrier emitter having a frequency of F_c. If the switch toggles between the two load values at a frequency F_{sw} then two subcarriers will appear in the received spectrum, one at frequency $F_c - F_{sw}$ and the other at $F_c + F_{sw}$ (Fig. 2.1).

Making the switch toggle between the loads at two distinct switching frequencies $F_{sw,0}, F_{sw,1}$, two sets of subcarriers will appear in the spectrum,

$$F_{sc,0} = F_c \pm F_{sw,0} \tag{2.2}$$

$$F_{sc,1} = F_c \pm F_{sw,1}. \tag{2.3}$$

The upper[1] (with respect to the main carrier (F_c) from the emitter) subcarriers of each set,

$$F_{sc-upper,0} = F_c + F_{sw,0} \tag{2.4}$$

$$F_{sc-upper,1} = F_c + F_{sw,1} \tag{2.5}$$

can be taken advantage as effectively, B-FSK modulation is realized: $F_{sc-upper,0}$ represents the binary value "0" and $F_{sc-upper,1}$ binary value "1". The time that each subcarrier is present in the air, defines the bit duration T_{bit} and thus the bit rate $R = \frac{1}{T_{bit}}$

By utilizing a circuit to control the switch in order to toggle between the two loads in two distinct frequencies, we can produce *high frequency* subcarriers and communicate with an appropriately tuned receiver. The circuit used to control the switch will be described in the next subsection (Subsec. 2.2.2).

[1]The lower subcarriers can be used as well, only in that case we need to invert the values we want to send. That is because when switching, for example, at $F_{sw,1}$ the lower subcarrier $F_{sc-lower,1}$ falls below the the lower subcarrier $F_{sc-lower,0}$ resulting from switching at $F_{sw,0}$ and if a receiver is tuned to listen around the center (as will be explained below) of these subcarriers it will receive the inverted version of the bit stream we want to send.

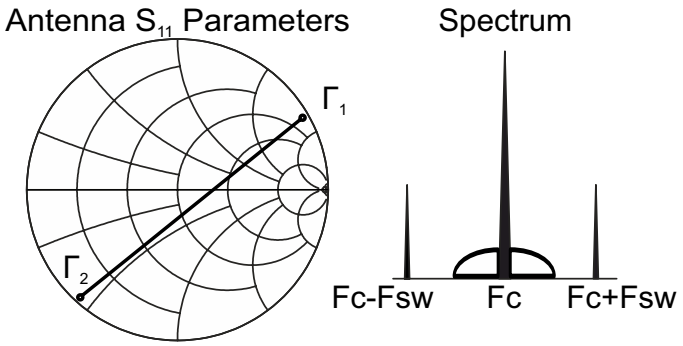


Figure 2.1: Generation of subcarriers due to scatter radio operation. Different antenna termination loads offer different reflection coefficients (left) that modulate a carrier signal with different amplitude and phase. For carrier frequency F_c, two subcarriers appear with frequencies $F_c \pm F_{sw}$.

2.2.2 Tag/Node Control Circuit

Control circuit-First Implementation

To test the concept, the first setup employed a Tektonix AFG3021B Arbitrary Waveform Generator (AWG) along with an Arduino Uno develoment (facilitating an Atmel ATmega 328 MCU running at 16MHz) board and the RF front end PCB with its antenna. A picture of the implementation can be seen in Fig. 2.2

The function generator was configured to produce 50% duty cycle square waves with $V_{pp} = 2V$ and $V_{min} = 0V$ in B-FSK Modulation mode. In that mode, 2 frequency values, $F_{sw,0}, F_{sw,1}$, are set in the generator. At any given time a square wave having one of those frequencies is present at the output. Which of the two frequencies will be present on the output and for how long, is controlled externally, in a binary fashion, by an arduino uno board. The output of the generator is connected to the RF front end.

A digital out pin (pin 7 was used) from the Arduino board is connected to the external modulation input of the generator through a resistor voltage divider network. The resistor divider is necessary due to the fact that the output levels of the arduino are $V_{low} = 0$ and $V_{high} = 5V$ and the generator input accepts levels at $V_{low} = 0$ and $V_{high} = 3.3V$. So the ratio of the divider must be,

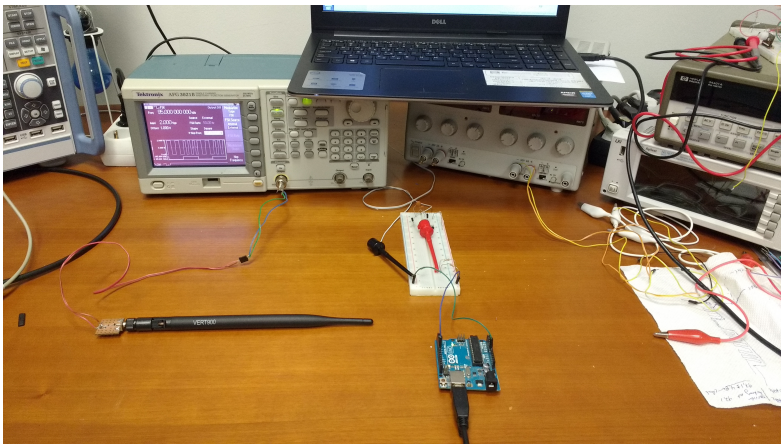


Figure 2.2: Setup of the first implementation. The arduino is powered by the laptop.

$$\frac{V_{in,generator}}{V_{out,arduino}} = \frac{3.3}{5} = 0.66 \tag{2.6}$$

Using the schematic in Fig. 2.3 the resistor divider equation can be written as,

$$\frac{V_{in,generator}}{V_{out,arduino}} = \frac{R_2}{R_1 + R_2} \tag{2.7}$$

Selecting $R_1 = 0.5$ kΩ and $R_2 = 1$ kΩ, and replacing these values in Eq. 2.7 we get $\frac{V_{in,generator}}{V_{out,arduino}} \simeq 0.66$ which is the required ratio from Eq. 2.6.

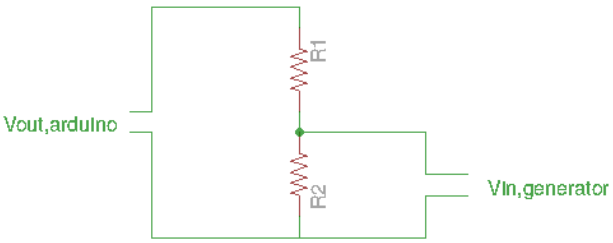


Figure 2.3: Resistor divider between generator's external modulation input and arduino output.

When the arduino's MCU sets the output pin High or Low, the generator outputs a square wave with frequency $F_{sw,1}$ or $F_{sw,0}$ respectively. The time

interval that the arduino's pin is held High or Low defines the duration that $F_{sw,1}$ or $F_{sw,0}$ will be present on the output.

The code that runs on the MCU of the arduino, contains a bit vector with the binary value represenation of the packet that the receiver is set to accept (Fig. 2.9). The MCU starts to read the vector value by value. If the current value is "1", the MCU will set the output pin High and will call a delay function, with an appropriate delay value as an argument, to hold the pin in that state for the necessary amount of time, thus maintaining the $F_{sw,1}$ signal at the output for that amount of time. After that, it will read the next value in the bit vector and will repeat the process. Accordingly, if the value is "0" the pin will be set to Low and the same procedure will follow.

In the generator, the two frequencies, $F_{sw,0}/F_{sw,1}$, can be adjusted in order for the resulting, due to scatter radio operation, subcarriers, to fall in a specific place in the frequency spectrum (Eq. 2.4, Eq. 2.5). The time interval that either of those frequencies is present on the output of the generator (and by extension driving the RF front end), was configured by the argument given in the delay function, used in the code running in the arduino, and was tuned using an oscilloscope connected to the output pin, in order for it to maintain a High/Low state lasting T_{bit}. A visual representation of the operation is shown in Fig. 2.4.

Control Circuit-Second Implementation

Having established the feasibility of the communication scheme using a function generator as the control circuit and the receiver described in Subsec. 2.2.4 , a more compact solution was employed in the second version of the control circuit. Instead of having the generator produce the switching frequencies and the arduino control their time duration and succession, the arduino was programmed so that it could do both. That way the test tag consisted only from the RF front end with its antenna and the arduino connected to that (Fig. 2.5 , Fig. 2.7).

In order for the arduino to produce the switching frequencies $F_{sw,0}, F_{sw,1}$, the same method described previously for generating the bit duration wave-

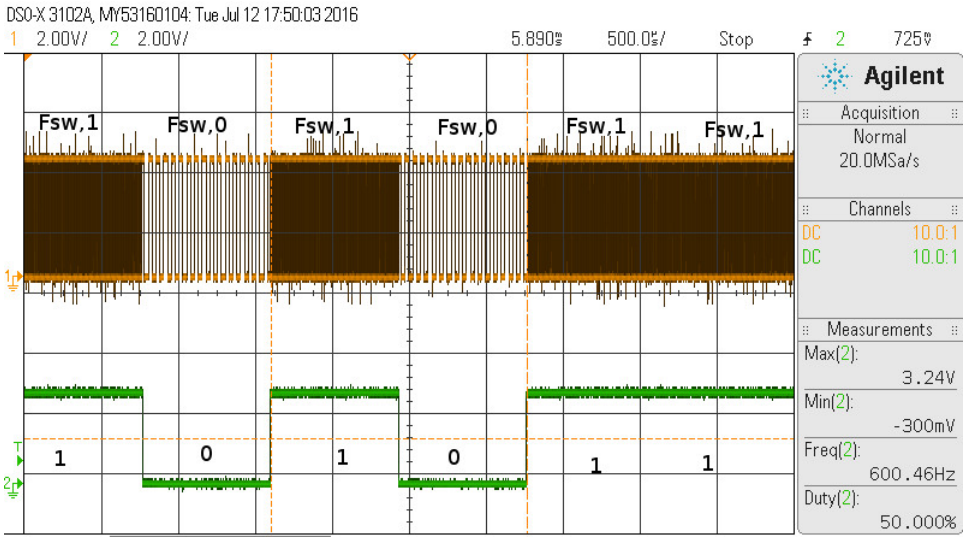


Figure 2.4: Output of the arduino (green trace) and output of the generator (orange trace). The switching frequencies are set to $F_{sw,0} = 50$ kHz, $F_{sw,1} = 100$ kHz for purposes of demonstrating the operation. Notice the Freq(2)$\simeq$ 600 Hz measurement, displaying the bit rate $R = 1200$ bps.

forms driving the generator, was used. The difference was the fact that the switching frequencies are much higher compared to the bit duration signal. Namely, the switching frequencies, as it will be explained in Subsec. 2.3.1, are in the order of 100 kHz and the bit duration signal, when there is an alternating 1/0 pattern, is , for a bit rate of $R = 1200$ bps, $\frac{R}{2} \simeq 600$ Hz.

Besides the relatively high (for the MCU) frequency values, the small

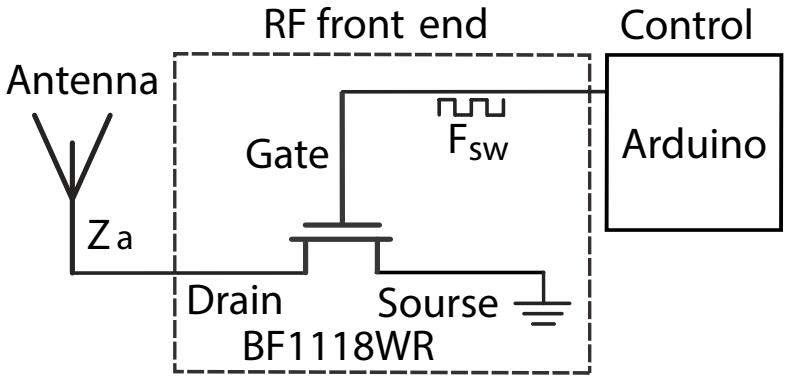


Figure 2.5: Block level view of the second implementation.

deviation is also a problem. The two switching signals, as it will be shown in Subsec. 2.3.1, must have frequencies whose values are close (their difference should give double the deviation i.e $\simeq$ 10 kHz). Using the method with the delay function to specify the duration a pin may maintain its state and then alter it, is not applicable in that case. That is because the timings are so tight that even the time the MCU needs to perform the call to the function affects them. Except that, as it will be explained later, to achieve a certain bit duration using the switching signals the repetition of them is needed and that repetition introduces delay.

To overcome the above problems, assembly No-Operation (NOP) instructions were used. To produce a signal with a specific frequency, a pin is set High. Then, a specific amount of NOP- calls is used. By running NOP instructions the program execution is effectively delayed thus making the pin maintain its current state for a time specified by the amount of NOPs[2] used. After these NOPs are executed the pin is set Low. Then a same amount of NOP instructions is executed to attain a duty cycle of 50% [3]. Running that code, will produce a one-shot square signal on the output pin. The time duration of that signal will be defined by the amount of NOP instructions and it has to be $T_{sw,i} = \frac{1}{F_{sw,i}},\ i \in \{0, 1\}$.

The amount of NOPs used in order to produce each of the frequencies was configured using the help of an oscilloscope. The code producing the one-shot square signal was run on a loop and by measuring the frequency of the resulting square waveform the amount was fine-tuned to attain the appropriate frequency value. The amount of NOPs before setting the pin Low and after was trimmed in order to achieve a duty cycle as close to 50% as possible.

To obtain a duration of T_{bit} in the resulting signals, each one-shot event having a time duration of $T_{sw,i},\ i \in \{0, 1\}$ must be repeated N_i times (Fig. 2.6).

[2]Usually a NOP instruction wastes a clock cycle in the MCU. In the case of the Arduino's uno setup the clock cycle is $T_{cycle} = \frac{1}{F_{clock}} = \frac{1}{16\ \text{MHz}} \simeq 62$ nsec. "Usually" means that it depends the MCU's core architecture.

[3]Driving the RF switch with a square wave having a duty cycle of 50%, elliminates the even number harmonics on the spectrum of the scattered signal.

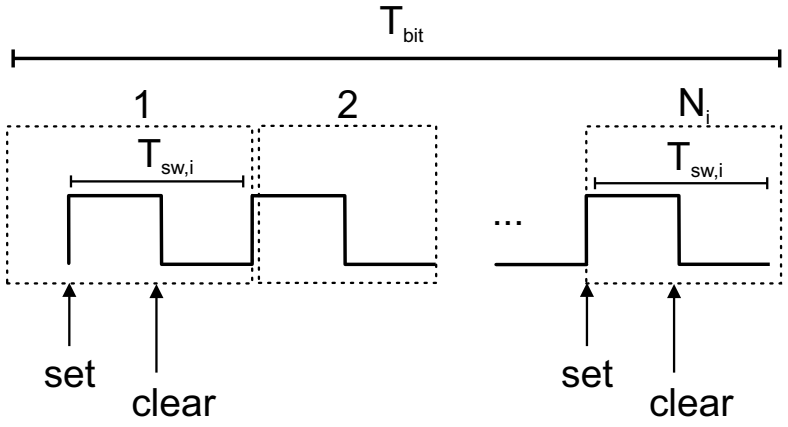


Figure 2.6: Period of subcarrier frequency and bit period; i = 0 for bit 0 and i = 1 for bit 1. Each one-shot signal can be seen inside the dotted lines.

So,

$$T_{bit} = N_i \cdot T_{sw,i} \Leftrightarrow N_i = \frac{T_{bit}}{T_{sw,i}}, i \in \{0, 1\} \tag{2.8}$$

The integer part of these values was modified by ±1, so that zero packet error rates were enjoyed at a high signal-to-noise ratio (i.e., small range between tag and receiver tested at the lab).

Just like the first implementation, a bit vector was used to represent the acceptable by the receiver packet. If the MCU read 1 in the current position of the bit vector, it called a function which executed the code used to generate a one shot signal with total duration of $T_{sw,1}$, N_1 times. Another similar function was called in order to generate the signal for 0.

2.2.3 RF front end-Switch

The RF switch used in this work is NXP's BF1118WR. This switch is a combination of depletion type Field Effect Transistor (FET) and a band switching diode. This transistor was chosen for this work due to the fact that it does not require external power, unlike other RF switching modules, and the low loss and high isolation capabilities that it offers. A visual representation of the switch can be seen in Fig. 2.5.

The switch was facilitated in a printed circuit board (PCB) (Chap. 6).

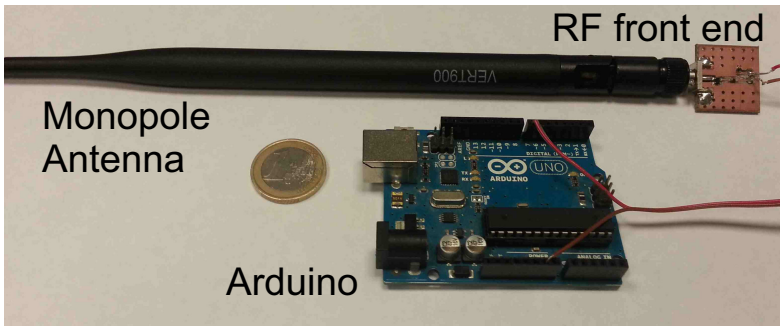


Figure 2.7: The Test Tag with the second control circuit implementation, consisting of an Arduino connected to the RF front end PCB and the monopole antenna.

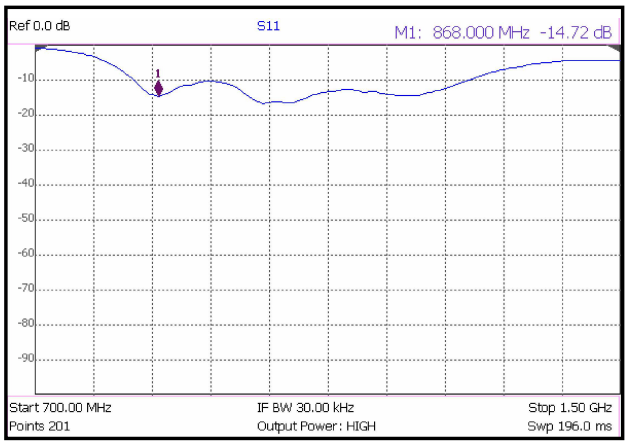


Figure 2.8: Measured reflection coefficient for the monopole antenna, (S_{11}), with marker at 868 MHz.

The antenna terminals were connected to Source and Drain terminals of the transistor. The transistor shorts (connects through a low resistance path) the terminals when $V_{sg} \geq 2V$ and open circuits them (lets them float) when $V_{sg} = 0V$. As can be seen in the schematic, the terminals of the transistor used to control it, are connected to the control circuit via emi suppression filters/beads to minimise the RF leak from the antenna.

The antenna used was a 3 dbi monopole. The measured S_{11} of the antenna at 868 MHz is offered at Fig. 2.8.

2.2.4 Embedded Receiver

The receiver module which was mainly used in this setup, is the Silicon Laboratories SI1064 ultra low power MCU with embedded transceiver, configured as receiver.

The selection of this specific integrated circuit (IC) was made mainly based on its sensitivity and because it was available in our laboratory. According to its datasheet, the module exhibits a typical sensitivity of -116 dbm with a receive filter bandwidth of 114 kHz, tuned at $450 - 470$ MHz, Gaussian-FSK (GFSK) modulation with BT=0.5, bit rate of 2.4 kbps and Bit Error Rate (BER) less than 0.1 %. The unit price of the IC is 3.89 Euros.

Receiver Configuration

Silicon laboratories provides a software tool called Wireless Development Suite (WDS) for configuring the SI1064. The tool performs automatic code generation depending on the desired radio configuration. The produced code is downloaded to the module and runs on the integrated MCU which in turn controls the radio sub system of the module. A Texas Instruments CC1101 sub-1-GHz Transceiver connected to Silicon Laboratories F320 MCU was also used in various steps of the implementation to double check the correct operation.

After configuring the necessary RF/Physical Layer and packet format related parameters, WDS produces code that can be compiled using tools provided in the Silicon Laboratories Integrated Development Environment (IDE). The default code produced by WDS just blinks some LEDs if it receives a valid packet. So to be able to see the received packets, and make measurements such as Packet Error Rate (PER), the code was modified to send the received packets either correct or erroneous, to the UART interface of the MCU. The development board that facilitates the module, has a UART to USB (virtual serial connection) converter and that way can be connected to a computer running a serial port terminal application and finally log the packets to a file.

Table 2.1: Receiver RF Configuration

Parameter	Value
Modulation	2-FSK
Data Rate	1.2 kbps
Center Frequency	868 MHz
Frequency Deviation	±5.157 kHz
RX Bandwidth	45 kHz

Physical Layer Configuration

The main configuration parameters for the receiver used in this link are presented on Table 2.1. Binary FSK modulation was selected due to its increased immunity to noise. In most wireless sensor networks there is no need for high speed communication, so a low bit rate of 1.2 kbps has been chosen for the proposed link. Having a low bit rate, frequency deviation can be small and thus the receiving filter bandwidth can be narrow.

Having a narrow filter at the receiver effectively lowers the noise floor. The spectral density of noise is given by,

$$N_0 = k \cdot T \text{ watt/Hz} \tag{2.9}$$

where $k = 1.38 \cdot 10^{-23}$ Joule/K the Boltzman's constant, and T the temperature in Kelvin. For a temperature of 27°C (300 K), 2.9 results:

$$10 \log N_0 = -174 \text{ dbm/Hz} \tag{2.10}$$

The value in 2.10, represents the power density of noise, measured in dbm per unit of bandwidth, at the previously mentioned temperature. If that value is multiplied by the bandwidth at which the receiver operates, i.e the bandwidth that is "allowed" by the receiver filter to pass to the rest of the signal processing chain in the system, we get the noise floor that the receiver "sees". This noise floor is further increased by a factor called Noise Figure (NF). Noise Figure is a measure of how much noise does the operation of the system itself, add to the the weak received signal. So the noise floor

seen by the receiver is:

$$P_{noise} = N_0 \cdot BW \cdot NF \tag{2.11}$$

Anything above P_{noise} level will be picked up by the receiver as useful signal. However, in many systems to achieve a certain "quality" (in digital communication the quality may be quantified with the Bit Error Rate (BER) metric) in the recovered signal, the input signal must be a little higher than P_{noise}. The amount of how much higher above the noise floor the signal must be, is defined as minimum Signal to Noise Ratio (SNR_{min}). If we are working with a digital communication system, SNR_{min} can be calculated using constraints regarding desired BER.

We can now define a quantity, that indicates the minimum power a signal may have, so that the receiver may "work on it" and produce an acceptable output (again, if we are talking about a digital communication scheme the acceptance of the output is judged by the BER). That quantity is called *Receiver Sensitivity* and it will be referred to as P_{min},

$$P_{min} = P_{noise} \cdot SNR_{min} \tag{2.12}$$

expressed in a log scale using dB,

$$P_{min} = \underbrace{10\log(N_0)}_{-174 \text{ dbm/Hz @ } 27^{\circ}\text{C}} + 10\log(BW) + 10\log(NF) + SNR_{min,dB} \tag{2.13}$$

Having mentioned the previous, it is obvious now that the less bandwidth a receiver is working on, the less power a signal induced in its antenna may have to be "acceptable" and so greater communication ranges can be achieved. In addition, having electronics that are optimized at hardware level (i.e integrated circuits) for implementing signal processing tasks at a specific receiver configuration, may result a better overall NF than a general purpose front-end utilized in a Software Defined Radio (SDR) system and thus further increasing the sensitivity.

Considering Table 2.1 and putting it in a very simplified manner, the receiver listens for signals around the center frequency of 868 MHz, if a carrier-signal falls at least 5.157 kHz (this value is called deviation) *below* or *above* the center frequency (and of course within the specified operating bandwidth), then the receiver interprets the presence of that carrier as an indication that a transmitting system has send the binary value of "0" or the binary value of "1" respectively.

The time that the carrier remains active defines the bit duration. So to send a binary value of "0", at a bit rate of $R = 1.2$ kbps which represents a bit duration of $T_{bit} = \frac{1}{R} = 833\ \mu sec$, we need to transmit a carrier at a frequency of $F_{sc,0} = 868$ MHz-5.157 kHz for a time duration of T_{bit} and hope that the receiver will hear us and recover the trasmitted "0". To transmit a "1" we follow the same procedure only this time we transmit at $F_{sc,1} = 868$ MHz $+ 5.157$ kHz .

Packet Format Configuration

Embedded radios like the one used, employ packet based communication. Except the data payload, the packet consists of a number of predefined bytes which are known to both the transmitter and the receiver. These are called preamble and synchronization bytes. There is the possibility of having extra bytes in the end of the packet that implement error detection using Cyclic Redundancy Check (CRC).

Preamble bytes is an alternating sequence of ones and zeros. This sequence helps the control loops used inside the receiver to perform bit clock recovery, frequency tracking/compensation, byte slicing etc. settle. Synchronization bytes are used, as the name suggests, for synchronization purposes i.e finding the beginning of the bytes carrying useful information inside the packet.

Sync-word and preamble bytes are user configurable, both in term of length in bytes and in content. For example, if a transmitter sends continuously packets with very little to zero time delay between each transmission, it's not necessary to have a long stream of preamble bytes because the con-

trol loops have settled. To minimize the chances of detecting noise as useful signal, a long stream of sync word bytes may be employed. Using more sync. word bytes, reduces the probability of noise appearing as a valid sync word and making the receiver think it received a valid packet structure although the received data bytes would probably be erroneous.

In the described link the packet had a length of 144 bits and consisted of:

1. Preamble bytes: A long preamble sequence of 8 bytes was chosen due to fact that there was a large time interval, namely 500 ms, between successive scatter radio "transmissions"

2. Synchronization word: A 4 byte sync. word was used. The default value was a two byte sync. word having value 0xD391. Using 4 bytes reduces the chances of detecting noise as useful signal (as mentioned earlier). The 4 byte pattern is a repetition of the 2 byte pattern.

3. Data payload: 6 data bytes were chosen because it was considered an appropriate amount of information for a WSN node.

The data payload was fixed to a value to perform the PER test described in Subsec. 2.3.2. A visual representation of the over-the-air packet can be seen in figure 2.9.

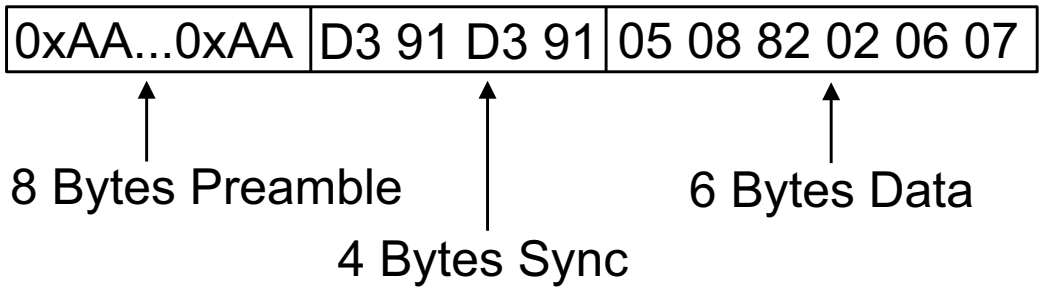


Figure 2.9: Packet Format.

2.3 Experimental Setup and Results

2.3.1 Experimental Setup

In this section the setup will be described, under which tests were performed to verify the operation as well as the performance of the link suggested in this work.

Selection of frequencies

In order for the receiver described in 2.2.4 and configured as in Table 2.1, to be able to receive the scatter radio signals, the upper side subcarriers must fall around the center frequency of $F_{center} = 868$ MHz. Specifically, taking into account the configured deviation, the subcarrier representing the value "0" must fall at,

$$F_{sc-upper,0} = 868 \text{ MHz} - 5.157 \text{ kHz} \simeq 867.995 \text{ MHz} \tag{2.14}$$

and in a similar way, the subcarrier representing "1" should end up at,

$$F_{sc-upper,1} = 868 \text{ MHz} + 5.157 \text{ kHz} \simeq 868.005 \text{ MHz} \tag{2.15}$$

Knowing where the subcarriers should be, the frequency of carrier emitter F_c must be chosen.

According to Eq. 2.4 and Eq. 2.5 , the closer the F_c is to the center frequency, the lower the switching frequencies need to be. Having a control circuit controlling the RF switch at low frequencies, makes the frequency generation a relatively easy task and most importantly lowers the power consumption as the later is proportional to the operating frequency, if we are talking for a digital system. But there is a problem having the F_c too close to F_{center} and that is *receiver desensitization*, the carrier emitter will appear as a strong interferer in the receiver and thus will degrade its performance.

In this work, the frequency of the carrier emitter was chosen to be at $F_c = 867.9$ MHz. At that frequency, the CW carrier is considered far enough, given the receive bandwidth of 45 kHz, from the receiver's center frequency.

Having the frequencies of both upper side subcarriers and the frequency of the illuminating carrier, it is easy to calculate from Eq. 2.4 and Eq. 2.5, the necessary switching frequencies. So for "transmitting" the bit "0" the control circuit has to make the RF switch toggle at,

$$F_{sw,0} = F_{sc-upper,0} - F_c = 867.995 \text{ MHz} - 867.900 \text{ MHz} \simeq 95 \text{ kHz} \quad (2.16)$$

and similarly using Eq. 2.5 we calculate $F_{sw,1}$ as,

$$F_{sw,1} = F_{sc-upper,1} - F_c = 868.005 \text{ MHz} - 867.900 \text{ MHz} \simeq 105 \text{ kHz} \quad (2.17)$$

To achieve a desired bit rate R, the control circuit must produce the appropriate switching signal for a duration of $T_{bit} = \frac{1}{R}$. For $R = 1.2$ kbps, $T_{bit} = 833$ μsec. That way, the subcarrier remains present for the necessary time interval in order for the specified bit rate to be achieved.

The whole operation described above to obtain B-FSK modulation, may be considered as an operation of a RF mixer/Up-converter, only in this case the parts consisting the mixer are dislocated. The RF input is the carrier emitter via the antenna, the Local Oscillator (LO) is the control circuit producing the switching frequencies and the scattered-transmitted signal is the RF output of the mixer.

Using an oscilloscope, a measurement of the switching frequencies was performed on the second control circuit implementation. The results were $F_{sw,0} = 94.06$ kHz and $F_{sw,1} = 104.52$ kHz. The switching frequency for the bit 1, may seem to fall slightly below the deviation limit, but the receiver copes well with it.

Fig. 2.10 shows an instance of the spectrum, produced by the second version of the control circuit, as seen by an appropriately configured Spectrum Analyzer.

In Fig. 2.10, M1 is the CW signal transmitted from the carrier emitter. M2, M3 are the subcarriers resulting from the switching operation of the

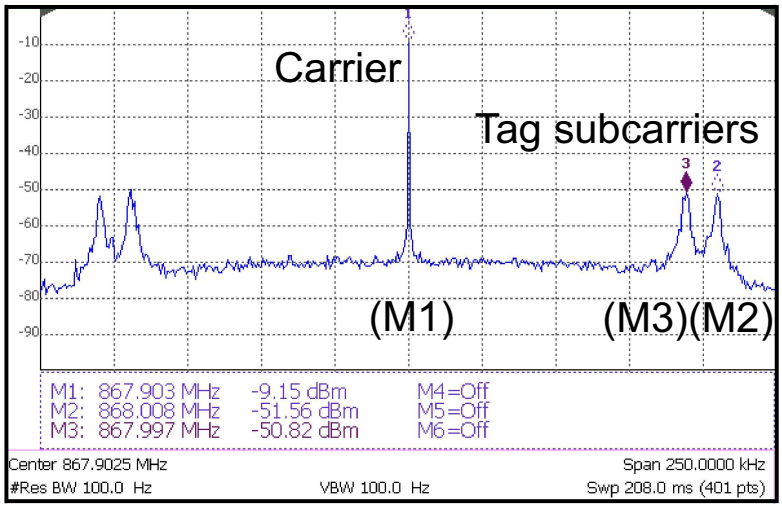


Figure 2.10: Received from a spectrum analyzer, spectrum of the scatter radio operation.

test tag, M2 resulting from switching at $F_{sw,1}$ and M3 from $F_{sw,0}$. One may observe the slight deviation of the carrier's frequency, from the desired $F_c = 867.9$ MHz by 3 kHz. This results a deviation of the subcarriers by the same amount but the receiver seems to cope well with it.

Carrier Emitter

As carrier emitter the same module, SI1064, used as receiver was employed. It was configured to transmit an unmodulated carrier wave at a frequency of F_c. The RF output power was configured to be 13 dbm. A photo of the system can be seen in Fig. 2.11.

To verify the operation of the link, the first experiments were done using the first implementation of the control circuit (Subsec. 2.2.2) configured with the aforementioned frequencies. These experiments were done under high-SNR conditions, i.e the distance between the carrier emitter and the tag and the distance between tag and the reader were in the order of a few meters. The tag transmitted packets every 1 sec and the receiver blinked a led if the received packets were correct.

Figure 2.11: Carrier emitter system.

2.3.2 Results

After verifying the operation of the link, it was time to test its performance. To test the performance of the communication link, the second control circuit implementation was used. A standard packet (Fig. 2.9) was transmitted every 500 msec and a total of 1000 packets were transmitted for each test. The SI1064 receiver forwarded the packets through a virtual serial port connection to a computer. The computer ran a serial port terminal program and logged the received packets into a file. When each test was over, the correct packets were counted and a PER value could be obtained. The setup was deployed in the outside garden of our department.

The PER-tests were performed with various d_{et}, d_{tr} combinations, where

- d_{et} : The distance between the emitter and the tag.
- d_{tr} : The distance between the tag and the reader.

and the results of those tests are shown in Table 2.2. It can be seen that 246 meters tag-to-reader distance is possible, with PER less than 1%, while

268 meters are possible at the expense of increased PER, in the order of 10%. The experimental setup utilized is vividly shown in Fig. 2.12.

Table 2.2: Communication Accuracy.

#	d_{et} (m)	d_{tr} (m)	PER (%)
1	3	69.7	0.1
2	3	145.4	0.1
3	3	246	0.9
4	3	268	10.6
5	8.4	40.7	15.5
6	25.2	24.8	1

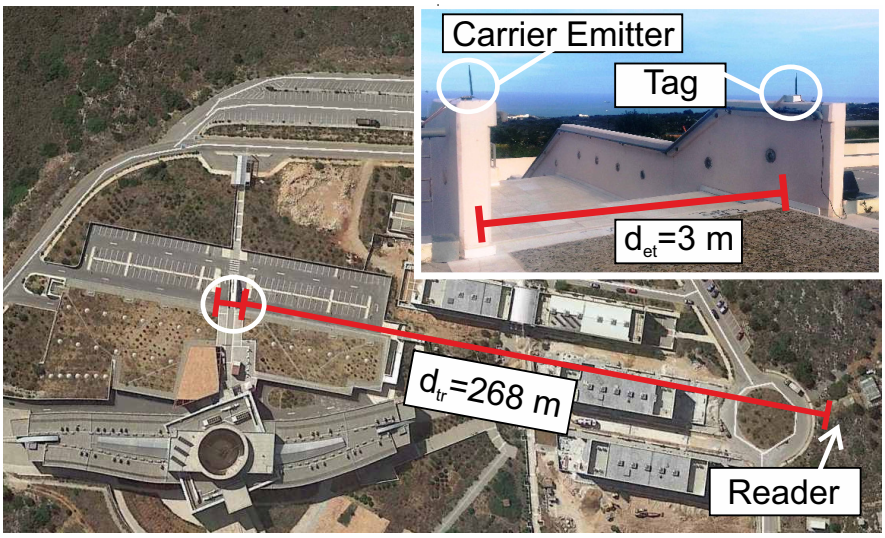


Figure 2.12: 268m test setup.

Chapter 3

Ambiently-Powered Scatter Radio Node

3.1 Introduction

The focus of this work was to implement a communication link, based on scatter radio principles, that would require a receiving system that could offer a high degree of integration and a low cost. From the tag's/node's perspective, the implementation of the link described in Chap. 2 was more of a proof of concept. It was made only for examining the feasibility of the concept and for testing its performance.

After assuring the correct operation of the link using the test tag, a prototype of a stand-alone node utilizing the functionality of the test tag was built. This prototype consisted of a Power Supply Unit to provide the system with power, an ultra low power MCU as a control circuit and an RF front end with its antenna.

The node's electronics were housed in the casing of a solar garden lamp (Fig. 3.1). These lamps operate during the night using a battery and turn off during the day for charging the battery via a solar panel. Using such a casing not only provides a nice package for the node, but offers a solar panel as well. These lamps cost 1-2 Euros each.

The total cost of the node was about 10 Euros, considering unit prices for the components and using expensive SMA connectors (which may be omitted if an integrated antenna/switch circuit is build). More details for the construction of the node will be given in the following sections.

Figure 3.1: Solar garden lamps used for casing. The left is the smaller one costing about 1.3 Euros and the right one is the bigger costing about 1.9 Euros.

3.2 Node Control Circuit and RF Front End

Just like the test tag in Subsec. 2.2.2, in order for the prototype node to communicate using scatter radio, a control circuit and a RF switch connected to an antenna are necessary.

3.2.1 RF Front End

The RF switch used in the node is the same used in the test tag (BF1118WR). The PCB (Chap. 6) facilitating the switch was redesigned in order to fit inside the solar garden lamp casing. A picture of the RF front end is shown in Fig. 3.2.

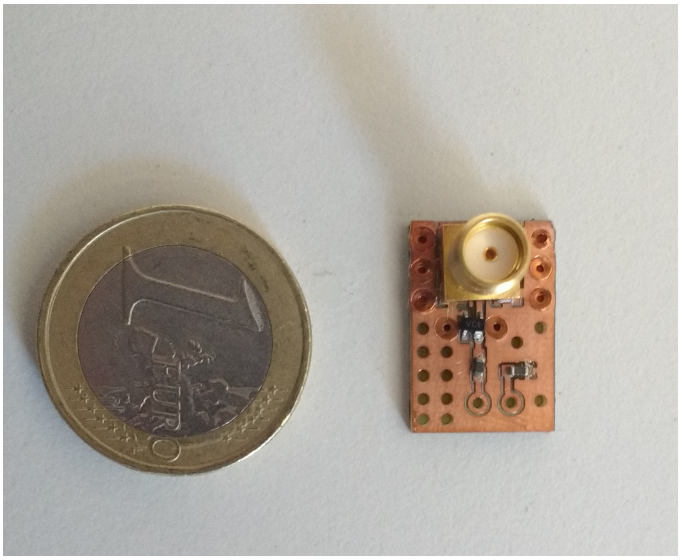


Figure 3.2: The prototype node's RF front end.

3.2.2 Control Circuit

The main component of the control circuit is the MCU. A Silicon Laboratories C8051F930 ultra low power MCU was employed. This MCU has an active current consumption of 170 μA/MHz. It has the capability of entering in a low power mode called "Sleep Mode" were the current consumption can be (according to datasheet) as low as 50 nA at a supply voltage of 1.8V. The

integrated 10 bit Analog to Digital Converter (ADC) can be used to take measurements from external sensors.

MCU operation

The MCU must be able to generate the switching signals with frequencies $F_{sw,0}, F_{sw,1}$ with the appropriate duration. To do this the same method used in the arduino of the second implementation of the test tag (Subsec. 2.2.2) was employed. The only difference was that C8051F930 was configured to run with a clock speed of $F_{clock} = 5$ MHz (in order to achieve a low active mode power consumption). So the amount of NOP instructions was different and of course the number N_i of one-shot signals needed to obtain a certain bit duration T_{bit} was different. The values were tuned, again, using an oscilloscope connected to the output driving the RF Front End.

The main operation of the MCU consists of taking 250, 8 bit voltage measurements using the ADC averaging them and then storing the average value in a variable. After that the main packet structure is formed. Packet structure is the same as that in Fig. 2.9 only in this case the data section is reduced to two bytes, one for the ADC value and the other is set to 0x82. Then, the appropriate signals are produced in order for scatter radio operation to be achieved.

In order to achieve ultra low power consumption the MCU was programmed to attain a duty cycled operation. During normal operation the microcontroller performs the tasks mentioned in the previous paragraph and then enters Sleep Mode. When in sleep mode, the MCU core, along with nearly all the peripherals are being disabled and the memory retains the data needed to continue normal operation from where is was stopped. The only peripheral that is chosen to be active during sleep mode is the Real Time Clock (RTC) module integrated in the MCU.

The RTC peripheral is able to count time using very low clock frequencies. The clock source of the RTC can be chosen either to be an external 32 kHz crystal or the internal RTC clock oscillator with adjustable frequency. In this case the internal RTC oscillator was chosen in order to avoid the cost

of an extra crystal. The default frequency of 20 kHz was used. The unit is able to produce interrupts that "wake up" the microcontroller from the sleep mode in time periods up to 36 hours. In case of our implementation the RTC was configured to produce interrupts in the order of 20 - 40 seconds. That amount can be configured simply by changing a value in the code.

The power consumption of the device was measured using an HP 34401 bench multimeter. To measure the active current consumption, the MCU was configured to stay continuously in active mode. The measured values were:

- Active Mode: $I_{active} = 2.37$ mA
- Sleep Mode: $I_{sleep} = 3.9$ μA

To calculate the average current consumption the duty cycle ratio, D, must be calculated. Duty cycle ratio can be given by,

$$D = \frac{t_{active}}{t_{sleep} + t_{active}} \tag{3.1}$$

where t_{active} is the time the MCU needs to perform the main operation and t_{sleep} the time that remains in sleep mode. To calculate t_{active}, a pin was set High when the MCU woke from the sleep mode and the same pin was set low, just before entering sleep mode. Using an oscilloscope (Fig. 3.3) the time duration of the pulse resulting from the previous operation was measured and was the t_{active} value. The measured value was $t_{active} = 115.2$ ms. To measure the specific t_{sleep}, the same pin was probed, only this time the oscilloscope configured to have a horizontal scale of 5 seconds/division, in order to see the time between successive changes of the pin's state. When configured to wake up every 25 seconds, the actual time that the MCU was in sleep mode was $t_{sleep} \simeq 17$ seconds[4]. Although the value is a bit off from the specified, it was consistent between successive wake-ups. Applying those values to Eq. 3.1 we get $D = \frac{115.2\text{ms}}{17115.2\text{ms}} = 0.0067$

[4]The reason this happens is believed to be the fact that the frequency of the RTC clock oscillator was left to the default value which is meant for sleeping intervals in the order of milliseconds. This frequency can be tuned down to 10 kHz.

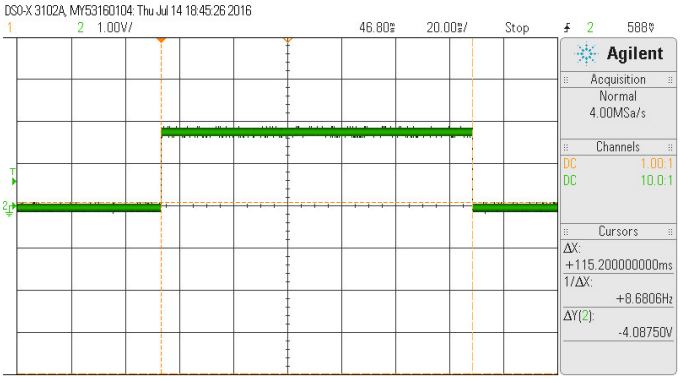


Figure 3.3: Measurement of t_{active}. The rise of the waveform indicated that the MCU woke up/entered active mode and the fall that it entered sleep mode. On the lower left the $\Delta X = t_{active} = 115.2$ ms of the cursors can be seen.

The average current can be calculated by,

$$I_{average} = D{\cdot}I_{active} + (1-D){\cdot}I_{sleep} = 0.0067{\cdot}2.37 \text{ mA } + 0.9932{\cdot}3.9\mu\text{A } \simeq 20\mu\text{A} \tag{3.2}$$

Of course we assume that the time that the device stays in sleep mode (t_{sleep}) is longer than t_{active}. It is obvious that the longer the device stays in sleep mode the lower the average current consumption will be.

3.3 Power Supply Unit

The solar garden lamps used to facilitate the electronics consist of a solar panel, a battery which charges from the solar panel during the day and a voltage boosting IC. The voltage boosting IC is needed because the battery used in the lamps is rated at 1.2V. At that voltage the LED is not capable of producing any light. So the voltage booster creates high voltage pulses and pulse-drives the LED but it does it fast enough that the human eye sees it like it's continuously emitting light. Although tests were done by modifying the circuit to continuously power the MCU and charge the battery, a different approach was selected using a more simple charging circuit along with a

supercapacitor instead of a battery.

3.3.1 Supercapacitor

In order to provide power to the system during the night, a supercapacitor was used. The constraint was to be able to power the system for about 10 hours. To estimate the time that a capacitor of a certain capacitance can power a system, we may use the relation between the current and voltage at a capacitors terminals,

$$I_{cap} = C \cdot \frac{\Delta V_{cap}}{\Delta t} \tag{3.3}$$

In our case the device is powered via a voltage reference with an output of 1.8V. We chose a voltage level of $V_{marginally-off} = 2\text{V}$ where we are close to having the device in a non operating state. The voltage rating of the capacitor used is $V_{max} = 2.7$ V and has a capacitance of $C = 35$ Farad. So, if we suppose that the capacitor is fully charged we have a $\Delta V_{cap} = V_{max} - V_{marginally-off} = 700$ mV. We can consider the current drawn by the supercapacitor to be the average current calculated in Eq. 3.2. So transforming Eq. 3.3 in order to give us the discharge time we get,

$$\Delta t_{discharge} = C \cdot \frac{\Delta V_{cap}}{I_{average}} = 35 \cdot \frac{700\text{mV}}{20\mu\text{A}} \simeq 340\ hours \tag{3.4}$$

The above equation is correct only if we are discharging the capacitor at a constant current, i.e the current flowing through the capacitor is DC. In our case, we approximate the switching current resulting from the duty cycled operation of the MCU with a DC current having a value of $I_{average}$, so the value is just an estimate. Besides that, the capacitor suffers from leakage currents and even if no load is connected to it, it will lose energy (and by extension the voltage will drop) over time. Except that, in the average current calculation we don't take into account the curents that are present during the wake up procedure which may have relatively large values.

The system was left to run for 13 hours and 29 minutes using a capacitor charged at $V_{cap} = 2.841\text{V}$ and having the solar panel covered. The measure-

Table 3.1: Capacitor voltage drop over time

Time	V_{cap}
12.39 PM	2.811 V
13.39 PM	2.750 V
14.39 PM	2.725 V
15.39 PM	2.706 V
16.39 PM	2.690 V
17.39 PM	2.676 V
18.39 PM	2.664 V
19.39 PM	2.653 V
20.39 PM	2.643 V
2.08 AM	2.595 V

ments begun at 12.39 PM and stoped at 2.08 AM. In Table 3.1 the measured values are shown.

Applying equation Eq.3.3 to the overall voltage change throughout the experiment,

$$I_{dc,equivalent} = C \cdot \frac{\Delta V_{cap}}{\Delta t} = 35 \text{ F} \cdot \frac{2.811 \text{ V} - 2.595 \text{ V}}{48540 \text{ sec}} \simeq 155\mu\text{A} \qquad (3.5)$$

This current is the DC current that would result such a drop of voltage in the time interval shown above. The calculated $I_{average}$ is almost an order of magnitude less than the equivalent DC calculated above. This may be explained by factors like the ones mentioned above.

As it can be seen in Table 3.1, the capacitor is able to power the circuit throughout a night and has a room for a further 595 mV of voltage drop.

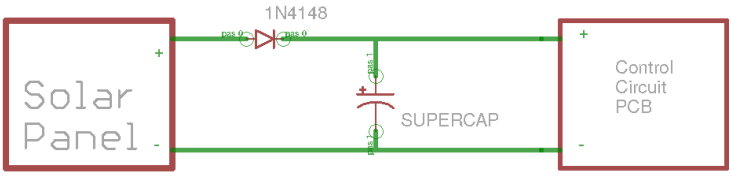


Figure 3.4: Power supply circuit utilizing a supercapacitor, a solar panel and a diode.

3.3.2 Charging System

Charging Circuit

The circuit employed in order to charge the supercapacitor, and in the same time provide power to the control circuit during the day, was a simple diode. The anode of the diode was connected to the (+) terminal of the solar panel and the cathode to the (+) terminal of the supercapacitor which at the same time was connected to the voltage input of the control circuit. The schematic of the circuit can be seen in Fig. 3.4. The diode provides isolation of the supercapacitor from the solar panel. That way, during the night current does not flow from the supercapacitor to the solar panel and energy is conserved. The drawback of having a diode is that the voltage at the supercapacitor will be a diode drop ($V_{drop} \simeq 600$ mV) lower than the voltage at the terminals of the solar panel thus not utilizing all the available energy.

Solar Panel

Two kinds of solar panels were tested (Fig. 3.5). Each one was taken from a different kind of solar lamp. The one coming from the small, in size, solar lamp had a open circuit voltage of $V_{oc,sm} = 3.4$V and a short-circuit current of $I_{sc,sm} \simeq 10$ mA. The one coming from the bigger lamp (that was also bigger in dimensions from the panel of the small lamp) had a $V_{oc,bg} = 2.3$V and a $I_{sc,bg} \simeq 20$ mA. These measurements were made using a portable multimeter at a full sunlight environment.

The bigger panel offers higher currents, thus allowing faster charge times according to Eq. 3.4. But the maximum voltage that the capacitor may reach is about $V_{max} = V_{oc,bg} - V_{drop} = 2.3 - 0.6 = 1.7$V, which is below the minimum

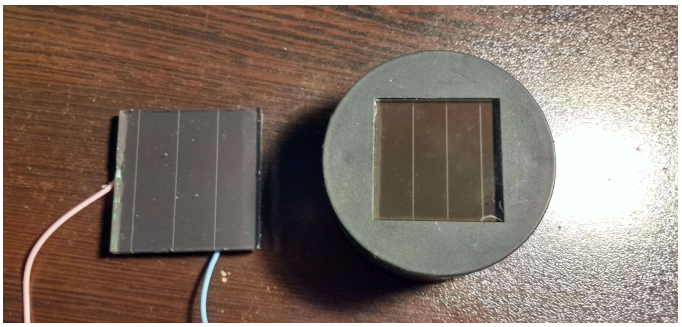


Figure 3.5: The solar panels taken from the garden lamps. On the left the bigger solar panel can be seen removed from its casing. On the right the smaller panel is shown in its casing.

operating voltage of the circuit ($1.9 - 2$ V). So the smaller solar panel was chosen having a $V_{max} = V_{oc,sm} - V_{drop} = 3.4 - 0.6 = 2.8$V.

The system was left for 1 hour to full sunlight environment (there were brief moments that the system was in shadow). The voltage of the capacitor before exposing the system to the sun was measured at $V_{cap,1} = 2.549$V. Then, the node was powered on and exposed to the sun. After an hour of operation, the capacitor voltage was measured again and had a value of $V_{cap,2} = 2.629$V, i.e the capacitor was charged by 80 mV. That indicated that the solar panel was able to charge the capacitor while the control circuit was operating. Although further testing is needed, it is believed that the panel will be able to fully charge the capacitor during the time duration of day. If it's not able, a panel with better specifications (a panel that is able to provide more current) may be employed.

3.4 Integration

The code that was written in order to perform the tasks described in Subsec. 3.2.2 , was initially tested on the C8051F930-DK development board in order to make sure of its correct operation. Standard values as the 2 byte data payload were chosen in the beginning to test the correct operation of the communication link. Then the 1st byte of the data payload was set to carry the ADC value and a laboratory power supply was connected to the

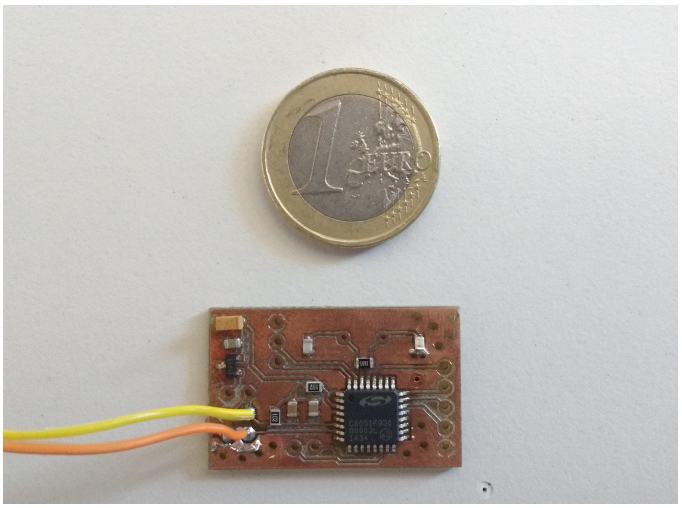


Figure 3.6: The PCB facilitating the MCU.

ADC's input to test the whole system. The receiving data were observed while the output of the power supply was changing to verify the operation of the whole system. The receiver used to test this setup was a Texas Instruments CC1101, configured like the SI1064 in Table 2.1, connected to a Silicon Laboratories C8051F320 development board. The board with the radio was connected to a computer via serial port and the received packets were dumped in a terminal screen.

After verifying the operation, a PCB (Fig. 3.6) was designed and fabricated to facilitate the microcontroller unit along with the components needed for its operation. These components include a Texas Instruments VREF3318 voltage reference Integrated Circuit (IC) (which in future designs may be omitted) as a voltage regulator providing a stable 1.8 output and other passive components. The complete system with the RF front end can be seen in Fig. 3.7.

The power supply unit was built in an prototyping board, along with a base for the supercapacitor, so that the last can be placed and removed easily.

Due to the size of the supercapacitor, the whole system could not be placed inside the smaller lamp, so the bigger was chosen. The problem was that the solar panel needed was that from the smaller lamp. So the solar

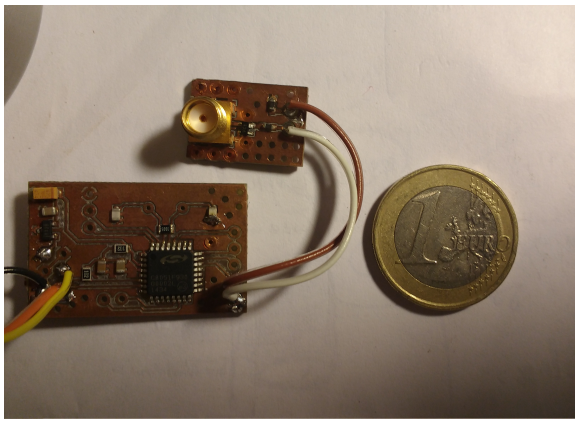


Figure 3.7: The Control Circuit PCB connected to the RF front end PCB.

panels were removed by both lamps using a multi function drilling tool and the smaller one was placed on the bigger lamp using hot glue and duct tape.

The solar panel was connected with the power supply board, and that board with the control circuit. The final node can be seen in Fig. 3.8 and Fig. 3.9.

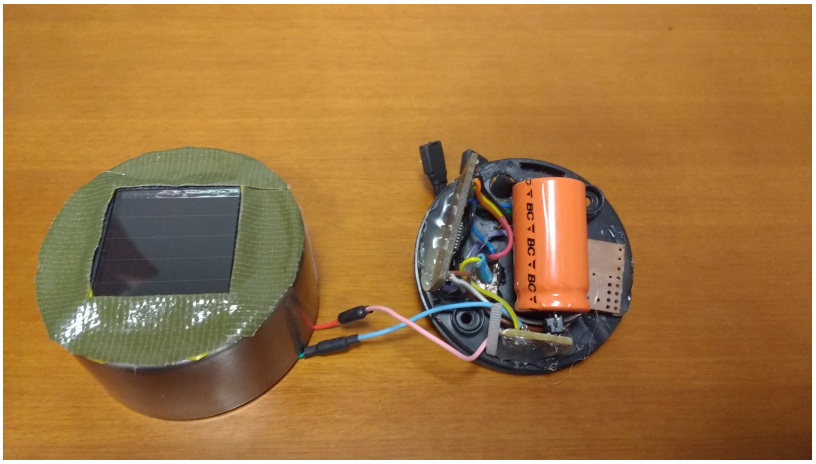


Figure 3.8: The node taken apart. On the left the solar panel can be seen. On the right the main electronics of the node are shown. The orange cylinder is the supercapacitor.

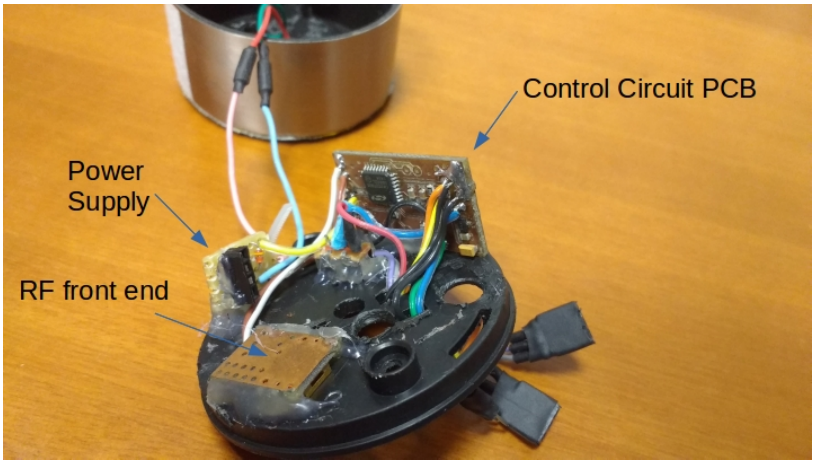


Figure 3.9: A close up of the node electronics. The supercapacitor is not present in the photo.

Chapter 4

Experimental Communication Link Using MCU Clock Harmonics

In this chapter, an experimental communication link, using only a Microcontoller Unit (MCU) and a conventional digital receiver module will be described. Although not related with scatter radio communication, it forms a way of communicating in short ranges, in the order of a couple of meters, using just a microcontroller unit and an appropriately tuned digital receiver module.

4.1 Description

Every MCU, being a complex digital circuit, needs a clock to run. The clock signal is ideally a square wave signal with, ideally, a 50% duty cycle. This signal is generated either with the help of an external crystal, or by a specific clock generator integrated circuit (IC) or internally, in the MCU itself, by dedicated internal oscillators.

The clock signal has to be "strong" enough. That is because it has to drive millions of transistors within the chip and at the same time maintain its signal integrity. To do so, the clock is generated by circuits capable of sourcing or sinking high values of currents. These currents may end up in current loops, thus radiating Electromagnetic Interference (EMI). EMI is the result of unintentional radiation caused by not well designed circuits which are driven by periodic current impulses and act as small antennas.

The frequency of the clock signal is configurable based on the application

needs. For most 8-bit microcontrollers the frequency is in the order of 10s of MHz. Being a square wave, the clock contains signals at integer multiples of its frequency.

In the following sections, we take advantage of the clock's harmonics and by driving the clock into to a long cable, acting as a small antenna, we achieve digital communication with an embedded receiver.

4.2 Transmitter

The transmitting part of the setup consisted of a Silicon Laboratories C8051F930 MCU in its development kit (Fig. 4.4, right). The MCU's clock was configured to be $F_{clock,conf} = 24.5$ MHz and was generated by the internal high frequency oscillator. The MCU used, has the capability of routing the clock signal into a specific pin, and this capability is achieved by writing a value in a specific register. By probing the pin with an oscilloscope we could measure the exact frequency of the clock signal which was $F_{clock} = 24.61$ MHz.

The code running in the MCU enables and disables the capability of the clock to be present on the pin. By performing such operation, the clock is present in the output in the form of bursts and thus we are effectively creating On-Off Keying (OOK) modulation with the carrier being the clock signal with frequency F_{clock}. In OOK the presence of the carrier denotes a binary value of "1" while the absence denotes the binary value of "0". The time that the carrier is present or absent defines the bit duration T_{bit} and by extension the bit rate $R = \frac{1}{T_{bit}}$.

In the MCUs code, a binary representation of the packet (Fig. 4.3) that the receiver expects to receive is created using a bit vector. The code reads the bit vector value by value. If the current value read is "1", the MCU enables the presence of the clock at the output. Then a delay function is called in order for a specific bit duration to be achieved by maintaining the carrier present at the output for a time of T_{bit}. Then the next value of the bit vector is read. If the value is "0" the same procedure is executed, only this time the carrier is absent from the output pin. The delay function is built using NOP instructions. The amount of delay is specified by how many

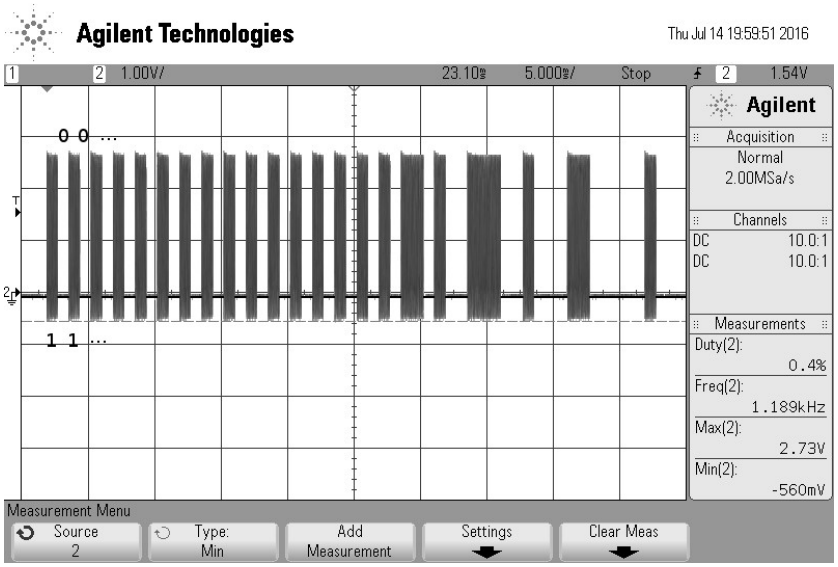


Figure 4.1: Oscilloscope capture of the output signal. Each burst represents the binary value "1". The absence of a 24.61 MHz burst represents a "0". One may observe that 6 bits occupy a time division of 5 ms $\Rightarrow T_{bit} = \frac{5000}{6} =$ 833 μsec wich represents a bit rate of $R = 1200$ bps.

of them will be executed in a loop. That amount was fine tuned using an oscilloscope to attain the necessary T_{bit}. In Fig. 4.1, the aforementioned operation is visually presented.

As mentioned earlier, the clock signal (ideally) being a square wave consists of signals at integer multiples of F_{clock}. This can be explained by expressing the square wave, using Fourier Series, as a sum of sine waves. Each sine wave consisting this sum, has a frequency which is an integer multiple of the square wave's frequency. This sine waves are called harmonics of the square wave. If the square wave has a 50% duty cycle, then only the sine waves having frequencies which are odd multiples of the square wave frequency are part of the sum. The amplitude, and therefore the power contribution, of each harmonic drops inversely proportional to its number.

In our case, the frequency of the square wave is $F_{clock} = 24.61$ MHz. A spectrum analyzer was used to see if the harmonics of that signal were "visible". The analyzer was tuned to different odd integer multiples of 24.61 MHz. The 13th harmonic was chosen for reasons that will be explained in

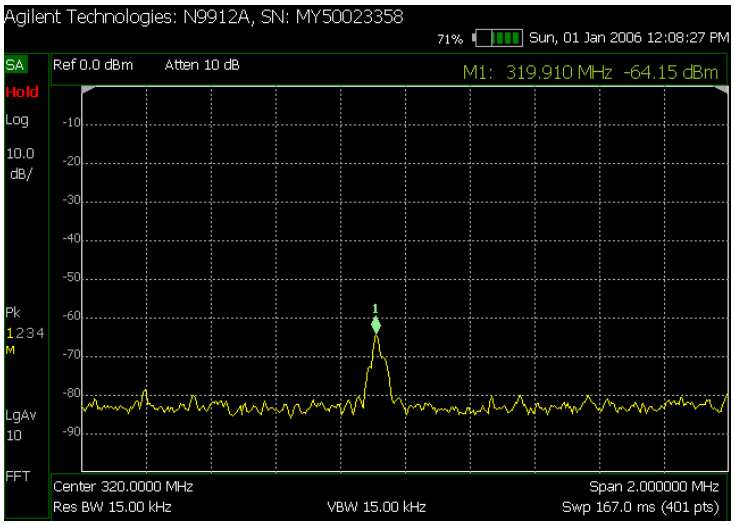


Figure 4.2: Spectrum around $F_{center,13}$. The transmitter/MCU was at a distance of 30 cm from the spectrum analyzer.

Sec. 4.3. The 13 th harmonic of F_{clock} is,

$$F_{center,13} = F_{clock} \cdot 13 = 24.61 \text{ MHz } \cdot 13 = 319.9 \text{ MHz} \tag{4.1}$$

The frequency spectrum around that frequency when the transmitter was operating can be seen in Fig. 4.2.

As an antenna, a simple breadboard hook cable was used connected to the clock output pin. The cable was about 10 cm in length.

4.3 Receiver

In order to receiver the OOK modulated information transmitted by the MCU, a Silicon Laboratories SI1064 module, like the one used in Subsec. 2.2.4, was used in its development kit.

4.3.1 Receiver Configuration

The receiver was configured using the WDS environment mentioned in Subsec. 2.2.4. After configuring the physical layer parameters, a standard packet

Table 4.1: Receiver RF Configuration

Parameter	Value
Modulation	OOK
Data Rate	1.2 kbps
Center Frequency	319.93 MHz
RX Bandwidth	200 kHz (Default Value)

format was set to the receiver, and if it received such a packet it blinked 2 LEDs. One LED indicated that it received a valid preamble/sync bit sequence, and the second LED indicated that the payload received was the same with the one configured, i.e a correct packet had arrived.

Physical Layer Configuration

The configuration parameters for the receiver used in this link are presented on Table 4.1. The receiver was configured to perform OOK demodulation. A bit rate of 1.2 kbps was chosen, higher bit rates can be achieved due to the fact that the MCU can perform smaller delays between toggling the clock output. The RX bandwidth was left to its default value. This choice was made because the MCU clock is not stable, so a considerable amount of frequency drift from the F_{clock} value is expected.

The SI1064's center frequency can be tuned within the ranges $283 - 350$ MHz, $425 - 525$ MHz and $850 - 960$ MHz. The 11th harmonic of F_{clock} is $24.61 \cdot 11 = 270.71$ MHz which is outside the lower operating range of the module. As shown in Eq. 4.1, the 13th is $F_{center,13} = 319.9$ which is within the operating range.

Packet Format Configuration

The configured standard packet had a length of 64 bits. It consisted of a 4 byte preamble stream, 2 byte synchronization word, and 2 data bytes as useful payload. A visual representation of the packet can be seen in Fig. 4.3.

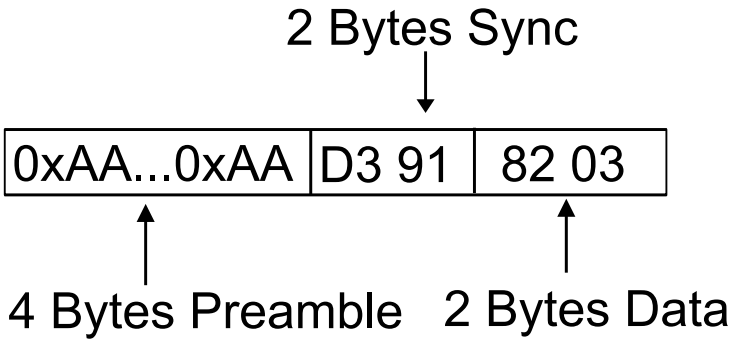


Figure 4.3: Packet configured to be accepted by the receiver.

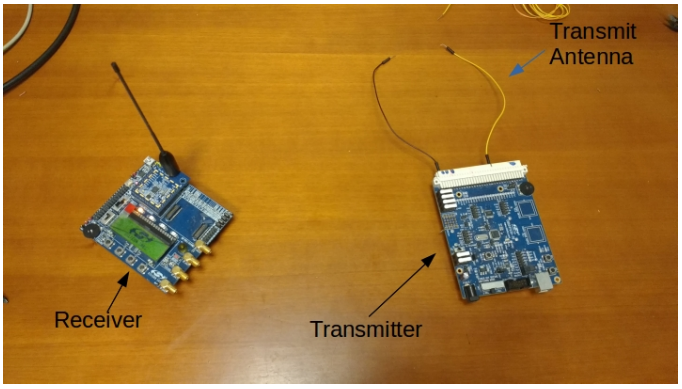


Figure 4.4: C8051F930 MCU in its Development kit as transmitter and the SI1064 embedded transceiver configured as receiver.

4.4 Test Setup

In order to briefly test the link, as mentioned earlier when the receiver had received a valid packet with correct data payload 2 LEDs blinked.

MCU was configured to transmit a packet every approximately 500ms. The receiver was moved away from the transmitter/MCU until the LED that by blinking indicated the reception of a packet with correct data payload, stopped blinking. That distance was in the order of 3 meters (indoors). Without the breadboard hook cable acting as an antenna, the transmitter had to be placed at a maximum distance of 10 cm from the receiver's antenna in order for that led to blink.

A photo of the setup is shown in Fig. 4.4.

Chapter 5

Conclusions

5.1 Conclusion

This work offered a concrete proof-of-concept of how existing low-cost, embedded radio, invented for conventional wireless sensor networks can be exploited to receive bistatic scatter radio signals, with ranges as high as 268 meters, at 13 dBm emitter power. Along with that a prototype node, powered by a supercapacitor and a solar panel was built, showing the possibility of implementing this communication scheme in a modern WSN. The communication link described in Chap. 4 can be exploited for short range communication needs.

5.2 Future Work

The main focus of this work was to implement a link by combining the benefits of bistatic scatter radio communication with the advantages of using conventional Marconi-type radio modules, commonly found in modern WSN implementations as readers.

The main aspect of the communication scheme presented that needs further investigation is the way a network may be constructed. The firsts thoughts for implementing a multiple access scheme to create a network are based on Time Domain Multiple Access (TDMA) scheme. The duty cycled operation may be exploited to offer random wake up patterns so each node wakes up and transmits its information when the others are sleeping. Because of the nature of the wake up system, the probability of a node waking up at the same time with another node, even if they are configured to wake up at the same time is very small.

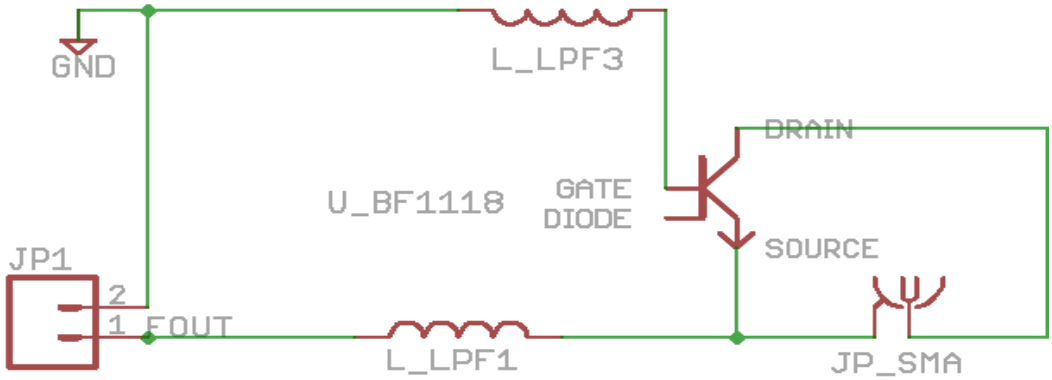
Another concept that is currently being investigated, is the use of ambient signals as illuminating carriers. Tests have been performed using signals from FM radio stations as carriers and the results are promising.

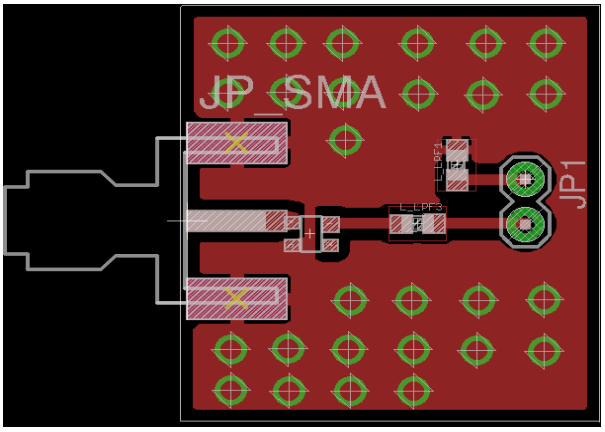
Chapter 6

Appendix

6.1 Schematics, PCBs & Bill of Materials

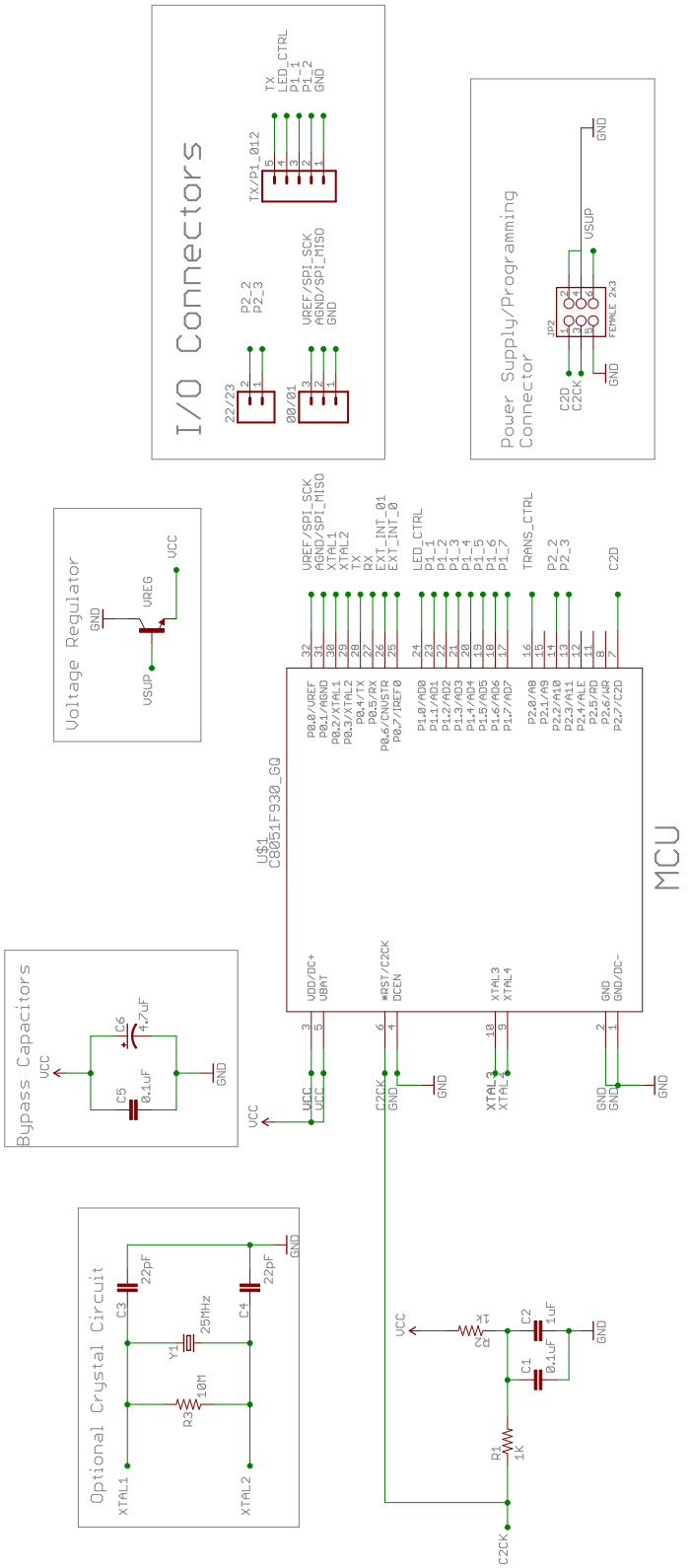
6.1.1 Test Tag RF front end



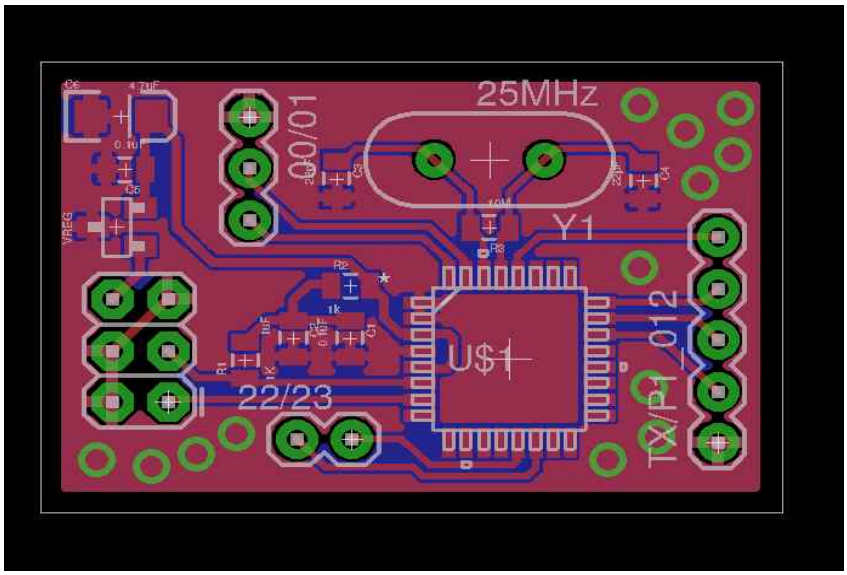


Part	MFR Code	MFR	Package	Description	Qty
L_LPF[1,3]	BLM18GG471SN1D	Murata Electronics	603	EMI bead	2
U_BF1118	BF1118WR115	NXP Semiconductors	SOT343R	RF switch	1
JP_SMA	132255	Amphenol RF	PCB-END LAUNCH	SMA Connector	1

6.1.2 Ambiently Powered-Scatter Radio Node Control Circuit Schematic



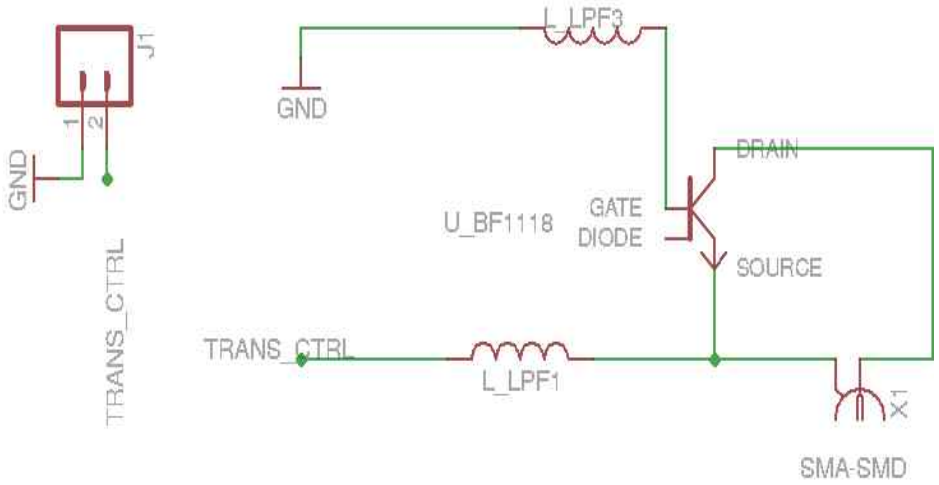
6.1.3 Ambiently Powered-Scatter Radio Node Control Circuit PCB & BOM

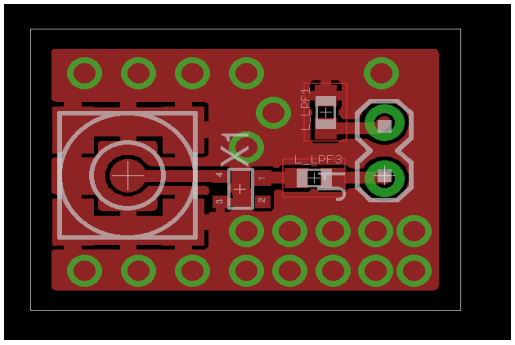


Part	MFR Code	MFR	Package	Description	Qty
C[1,5]			805	0.1 uF Capacitor	2
C2			805	1 uF Capacitor	1
C6			805	4.7 uF Capacitor	1
R[1,2]			805	1k Resistor	2
U1	C8051F930-GQ	Silicon Laboratories	QFP32	MCU	1
VREG	VREF3318	Texas Instruments	SOT23	1.8V Voltage Reference	1

Crystal is not used, so the related components are omitted.

6.1.4 Ambiently Powered-Scatter Radio Node RF front end





Part	MFR Code	MFR	Package	Description	Qty
L_LPF[1,3]	BLM18GG471SN1D	Murata Electronics	603	EMI bead	2
U_BF1118	BF1118WR115	NXP Semiconductors	SOT343R	RF switch	1
JP_SMA	142-0711-201	Johnson / Cinch Connectivity Solutions	VERTICAL STRAIGHT SMA	SMA Connector	1

Bibliography

[1] D. Evans. (2011) The Internet of Things. How the next evolution of the internet is changing everything. [Online]. Available: http://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[2] H. Stockman, "Communication by means of reflected power," *Proc. IRE*, pp. 1196–1204, 1948.

[3] E. Kampianakis, J. Kimionis, K. Tountas, C. Konstantopoulos, E. Koutroulis, and A. Bletsas, "Wireless environmental sensor networking with analog scatter radio and timer principles," *IEEE Sensors J.*, vol. 14, no. 10, pp. 3365–3376, 2014.

[4] S. N. Daskalakis, S. D. Assimonis, L. Kampianakis, and A. Bletsas, "Soil moisture wireless sensing with analog scatter radio, low power, ultra-low cost and extended communication ranges," in *Proc. of IEEE Sensors Conf.*, Valencia, Spain, Nov. 2014.

[5] A. Dementyev and J. R. Smith, "A wearable UHF RFID-based EEG system," in *Proc. IEEE Int. Conf. on RFID*, 2013, pp. 1–7.

[6] J. Kimionis, A. Bletsas, and J. N. Sahalos, "Design and implementation of RFID systems with software defined radio," in *6th IEEE European Conf. on Antennas and Propagation (EuCAP)*, Prague, Czech Republic, Mar. 2012, pp. 3464–3468.

[7] ——, "Bistatic backscatter radio for power-limited sensor networks," in *Proc. IEEE Global Telecommunications Conference (Globecom)*, Atlanta, GA, Dec. 2013, pp. 375–380.

[8] ——, "Increased range bistatic scatter radio," *IEEE Trans. Commun.*, vol. 62, no. 3, pp. 1091–1104, Mar. 2014.

[9] N. Fasarakis-Hilliard, P. Alevizos, and A. Bletsas, "Coherent detection and channel coding for bistatic scatter radio sensor networking," *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1798–1810, May 2015.

[10] J. F. Ensworth and M. S. Reynolds, "Every smart phone is a backscatter reader: Modulated backscatter compatibility with bluetooth 4.0 low energy (ble) devices," in *Proc. IEEE Int. Conf. on RFID*, San Diego, CA, Apr. 2015.

[11] G. Vougioukas, S. N. Daskalakis, and A. Bletsas, "Could battery-less scatter radio tags achieve 270-meter range?" in *2016 IEEE Wireless Power Transfer Conference (WPTC)*, Aveiro, Portugal, May 2016, pp. 1–3.

[12] B. Razavi, *RF Microelectronics*. New Jersey: Prentice-Hall, 1998.