Two-Way Digital Paging System Using Software Defined Radios

Ratnayake R.M.S.H.(230548R)
Tennakoon U.G.R.B.(230629R)
Disssanayake R.K.T.(230164K)
Shehan M.N.N.(230613M)

Department of Electronic & Telecommunication Engineering
University of Moratuwa, Sri Lanka

September 17, 2025

Presentation Structure

1. Introduction

2. Transmitter

3. Receiver

4. Selected Methodologies

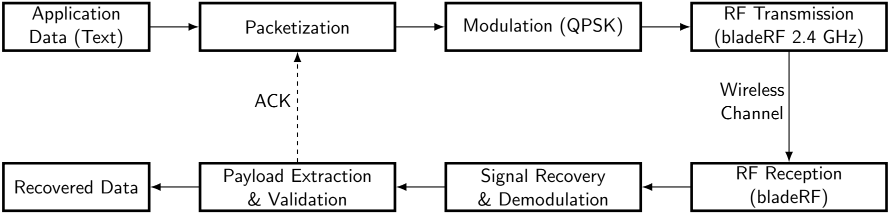
Abstraction

This project demonstrates the design of a **two-way digital paging system** using **GNU Radio** and **BladeRF software-defined radios**. The system ensures **reliable wireless messaging** through QPSK modulation, addressing, framing, and acknowledgment mechanisms. **CRC-based error detection** enhances data integrity, while a real-time GUI visualizes transmission and reception. The prototype showcases the **practical power of SDR platforms** in building robust and extensible digital communication systems.

Requirements

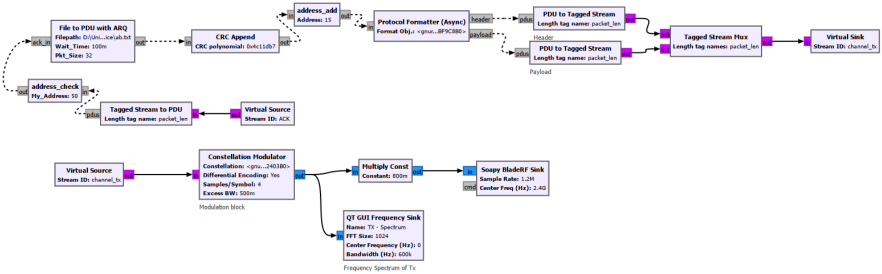
- Short message delivery using digital modulation (BPSK/QPSK).
- Unique user addressing (receiver responds only to its ID).
- Acknowledgment (ACK) mechanism for reliable communication.
- CRC-based error detection and rejection of corrupted messages.
- Basic user interface (console or GUI) for composing messages.

Block Diagram



Transmitter

Message Transmitter Block



File to PDU with ARQ
Filepath: D:\Uni...ice\ab.txt
Wait_Time: 100m
Pkt_Size: 32

CRC Append
CRC polynomial: 0x4c11db7

address_add
Address: 15

Protocol Formatter (Async)
Format Obj.: <gnur...BF9C8B0>

PDU to Tagged Stream
Length tag name: packet_len

Header

PDU to Tagged Stream
Length tag name: packet_len

Payload

Tagged Stream Mux
Length tag names: packet_len

Virtual Sink
Stream ID: channel_tx

address_check
My_Address: 50

Tagged Stream to PDU
Length tag name: packet_len

Virtual Source
Stream ID: ACK

Virtual Source
Stream ID: channel_tx

Constellation Modulator
Constellation: <gnu...2403B0>
Differential Encoding: Yes
Samples/Symbol: 4
Excess BW: 500m

Modulation block

Multiply Const
Constant: 800m

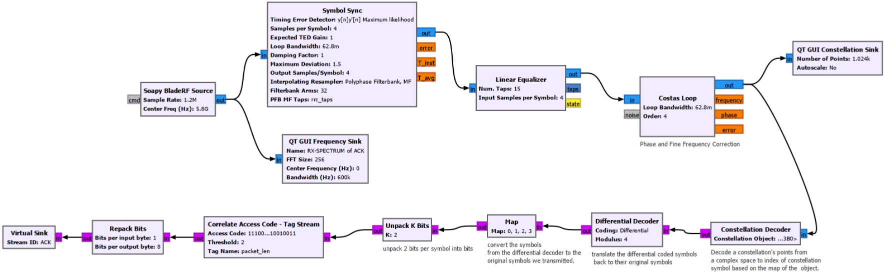
Soapy BladeRF Sink
Sample Rate: 1.2M
Center Freq (Hz): 2.4G

QT GUI Frequency Sink
Name: TX - Spectrum
FFT Size: 1024
Center Frequency (Hz): 0
Bandwidth (Hz): 600k

Frequency Spectrum of Tx

Figure: Message Transmitter Block

Acknowledgement Receiver Block


Soapy BladeRF Source
Sample Rate: 1.2M
Center Freq (Hz): 5.8G

Symbol Sync
Timing Error Detector: y[n]y'[n] Maximum likelihood
Samples per Symbol: 4
Expected TED Gain: 1
Loop Bandwidth: 62.8m
Damping Factor: 1
Maximum Deviation: 1.5
Output Samples/Symbol: 4
Interpolating Resampler: Polyphase Filterbank, MF
Filterbank Arms: 32
PFB MF Taps: rrc_taps

QT GUI Frequency Sink
Name: RX-SPECTRUM of ACK
FFT Size: 256
Center Frequency (Hz): 0
Bandwidth (Hz): 600k

Linear Equalizer
Num. Taps: 15
Input Samples per Symbol: 4

Costas Loop
Loop Bandwidth: 62.8m
Order: 4

Phase and Fine Frequency Correction

QT GUI Constellation Sink
Number of Points: 1.024k
Autoscale: No

Constellation Decoder
Constellation Object: ...3B0>

Decode a constellation's points from
a complex space to index of constellation
symbol based on the map of the object.

Differential Decoder
Coding: Differential
Modulus: 4

translate the differential coded symbols
back to their original symbols

Map
Map: 0, 1, 2, 3

convert the symbols
from the differential decoder to the
original symbols we transmitted.

Unpack K Bits
K: 2

unpack 2 bits per symbol into bits

Correlate Access Code - Tag Stream
Access Code: 11100...10010011
Threshold: 2
Tag Name: packet_len

Repack Bits
Bits per input byte: 1
Bits per output byte: 8

Virtual Sink
Stream ID: ACK

Figure: Acknowledgement Receiver Block

Blocks & Descriptions

Message Transmitter Block

- **Message Source:**[1] Custom Python block that reads the input file/message, segments it into packets, and prepares data for transmission. It also processes acknowledgment (ACK) messages from the receiver.
- **CRC Append:** Adds a Cyclic Redundancy Check (CRC) code to each packet, enabling error detection at the receiver and ensuring corrupted messages are discarded.
- **Protocol Formatter:** Frames each packet with a header (containing sync word, addressing, etc.) and payload, ensuring proper synchronization and identification.
- **Address Add:**[1] Add the destination address to the frame

[1] Custom Python Block

Blocks & Descriptions (Contd...)

- **PDU ↔ Tagged Stream Conversion:** Converts packet-based data (PDUs) into tagged streams for modulation, and back again at the receiver. This allows GNU Radio blocks to handle both continuous streams and discrete packets.
- **Stream Mux:** Combines headers and payloads into a single stream for transmission.
- **QPSK Modulator:** Maps digital bits into complex QPSK symbols for RF transmission, providing bandwidth efficiency and robustness.

Acknowledgement Receiver Block

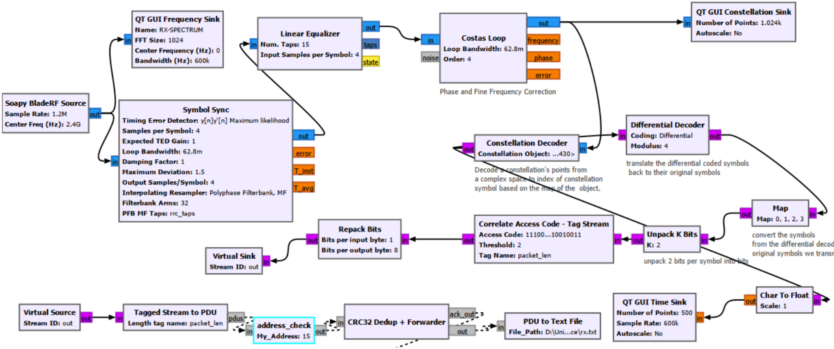
- **Symbol Synchronization:** Aligns the received signal samples with symbol timing to reduce inter-symbol interference.
- **Linear Equalizer:** Compensates for channel distortions and multipath effects, improving signal quality.

Blocks & Descriptions (Contd...)

- **Costas Loop:** Corrects carrier frequency and phase offsets in the received signal, enabling proper demodulation.
- **QPSK Decoder:** Converts received QPSK symbols back into digital bits.
- **Differential Decoder:** Removes phase ambiguity introduced during modulation/demodulation.
- **Bit Mapping + Unpack/Repack:** Reconstructs the bitstream into meaningful packets, ready for higher-layer processing.
- **Address Check:**[1] Checking whether the destination address is correct and passing though it.
- **ACK Feedback Path:**[1] Ensures reliable delivery by informing the transmitter whether a message was correctly received.

Receiver

Message Receiver Block



Soapy BladeRF Source
Sample Rate: 1.2M
Center Freq (Hz): 2.4G

QT GUI Frequency Sink
Name: RX-SPECTRUM
FFT Size: 1024
Center Frequency (Hz): 0
Bandwidth (Hz): 600k

Symbol Sync
Timing Error Detector: y[n]y'[n] Maximum likelihood
Samples per Symbol: 4
Expected TED Gain: 1
Loop Bandwidth: 62.8m
Damping Factor: 1
Maximum Deviation: 1.5
Output Samples/Symbol: 4
Interpolating Resampler: Polyphase Filterbank, MF
Filterbank Arms: 32
PFB MF Taps: rrc_taps

Linear Equalizer
Num. Taps: 15
Input Samples per Symbol: 4

Costas Loop
Loop Bandwidth: 62.8m
Order: 4

Phase and Fine Frequency Correction

QT GUI Constellation Sink
Number of Points: 1.024k
Autoscale: No

Constellation Decoder
Constellation Object: ...430>

Decode a constellation's points from
a complex space to index of constellation
symbol based on the map of the object.

Differential Decoder
Coding: Differential
Modulus: 4

translate the differential coded symbols
back to their original symbols

Map
Map: 0, 1, 2, 3

convert the symbols
from the differential decoder to the
original symbols we transmitted.

Unpack K Bits
K: 2

unpack 2 bits per symbol into bits

Correlate Access Code - Tag Stream
Access Code: 11100...10010011
Threshold: 2
Tag Name: packet_len

Repack Bits
Bits per input byte: 1
Bits per output byte: 8

Virtual Sink
Stream ID: out

Char To Float
Scale: 1

QT GUI Time Sink
Number of Points: 500
Sample Rate: 600k
Autoscale: No

Virtual Source
Stream ID: out

Tagged Stream to PDU
Length tag name: packet_len

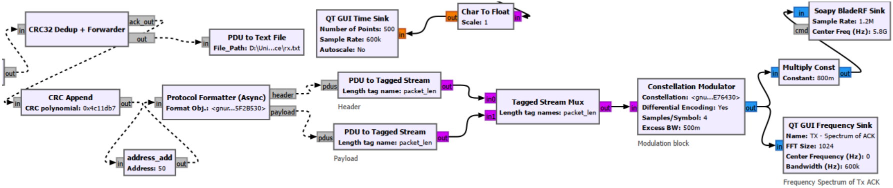
address_check
My_Address: 15

CRC32 Dedup + Forwarder

PDU to Text File
File_Path: D:\Uni...ce\rx.txt

Figure: Message Receiver Block Diagram

Acknowledgement Transmitter Block



CRC32 Dedup + Forwarder

PDU to Text File
File_Path: D:\Uni...ce\rx.txt

QT GUI Time Sink
Number of Points: 500
Sample Rate: 600k
Autoscale: No

Char To Float
Scale: 1

Soapy BladeRF Sink
Sample Rate: 1.2M
Center Freq (Hz): 5.8G

CRC Append
CRC polynomial: 0x4c11db7

Protocol Formatter (Async)
Format Obj.: <gnur...5F2B530>

PDU to Tagged Stream
Length tag name: packet_len

PDU to Tagged Stream
Length tag name: packet_len

address_add
Address: 50

Tagged Stream Mux
Length tag names: packet_len

Constellation Modulator
Constellation: <gnu...E76430>
Differential Encoding: Yes
Samples/Symbol: 4
Excess BW: 500m

Multiply Const
Constant: 800m

QT GUI Frequency Sink
Name: TX - Spectrum of ACK
FFT Size: 1024
Center Frequency (Hz): 0
Bandwidth (Hz): 600k

Figure: Acknowledgement Transmitter Block

Other than blocks in transmitter, we only used,

- **CRC 32 Dedup + Forwarder:**[1] Receives packets, checks their CRC32 for integrity, forwards the payload only if it's not a duplicate, and always sends an acknowledgment (ACK) for valid packets. Essentially, it acts as a CRC validator, deduplicator, and forwarder.

Selected Methodologies

Frame structure

Preamble (4 B)	DST Address (1 B)	Sq ID + Payload ($\leq$ 32 B)	CRC 32 (4 B)

Figure: Message Frame

Preamble (4 B)	DST Address (1 B)	0xAA + Sq ID (2 B)	CRC 32 (4 B)

Figure: Acknowledgement Frame

Maximum text file size[2] $=(2^8 - 1) \times N$ Bytes, where N is the number of data bytes in payload.

[2] For this frame design $N = 31$. Therefore, Maximum text file size = 7905 Bytes.

ARQ Mechanism

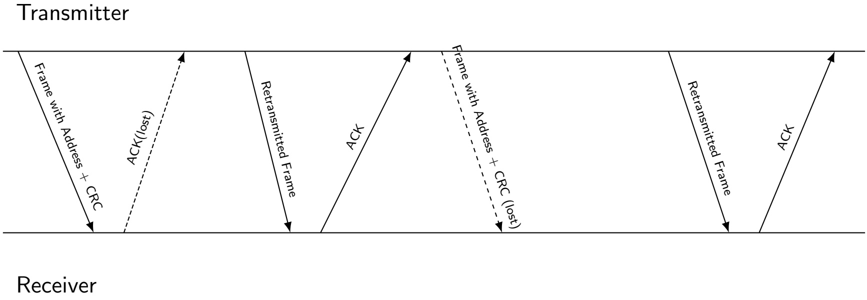
We are using a **Stop-and-Wait ARQ** system:

Step 1: Each frame carries an **address**.

Step 2: The receiver checks the **address**.

Step 3: If correct, it sends an **ACK** back.

Step 4: The sender waits for the ACK. If it is missing, the sender **retransmits** the same frame.



How ARQ Mechanism Visible in GNU Radio?

```
[CRC32] OK (Packet 20)
[Forward] Packet 20 duplicate, not forwarded
[FilePDU] Timeout waiting for ACK of 0x15, retrying...
[FilePDU] Packet 0x15 sent (try 2)
[CRC32] OK (Packet 21)
[Forward] UTF-8 decode error: 'charmap' codec can't encode character '\u2192' in
position 20: character maps to <undefined>
[TextFile] Wrote 31 chars to D:\University\Semester 3\EN2130\gnupractice\rx.txt
[FilePDU] Timeout waiting for ACK of 0x15, retrying...
[FilePDU] Packet 0x15 sent (try 3)
[CRC32] OK (Packet 21)
[Forward] Packet 21 duplicate, not forwarded
[FilePDU] ACK received for packet 0x15
[FilePDU] Packet 0x16 sent (try 1)
[CRC32] OK (Packet 21)
[Forward] Packet 21 duplicate, not forwarded
[FilePDU] Timeout waiting for ACK of 0x16, retrying...
[FilePDU] Packet 0x16 sent (try 2)
[CRC32] OK (Packet 22)
[Forward] UTF-8 decode error: 'charmap' codec can't encode character '\u2192' in
position 20: character maps to <undefined>
[TextFile] Wrote 7 chars to D:\University\Semester 3\EN2130\gnupractice\rx.txt
[FilePDU] Timeout waiting for ACK of 0x16, retrying...
[FilePDU] Packet 0x16 sent (try 3)
[CRC32] OK (Packet 22)
[Forward] Packet 22 duplicate, not forwarded
[FilePDU] ACK received for packet 0x16
```

QPSK Modulation

- **Quadrature Phase Shift Keying (QPSK)** is used as the modulation scheme.
- Each symbol carries **2 bits**, mapped to one of four constellation points.
- Constellation points: $(\pm1, \pm1)$ with Gray coding to minimize bit errors.
- **Advantages:**
  - Bandwidth efficient (2 bits/symbol).
  - Robust against noise compared to higher-order modulations.
- Symbol synchronization is used to correctly recover symbols at the receiver.

Simulation Results

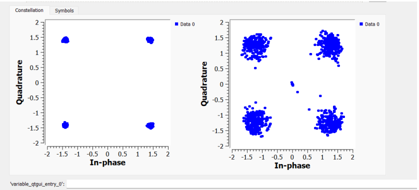


Figure: Constellation Diagrams

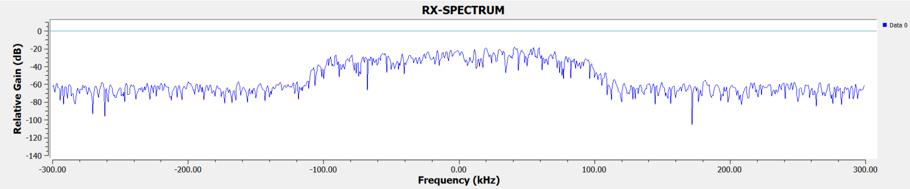


Figure: Frequency Spectrum

References

GNU Radio, “Tutorials,” GNU Radio Wiki. [Online]. Available: `https://wiki.gnuradio.org/index.php/Tutorials`. [Accessed: Sep. 14, 2025].

Thank You!