

Linear Block Coding

Purvil Kamdar

Computer Engineering Department, College of Engineering
San Jose State University, San Jose, CA 95192
E-mail: purvilkamdar92@gmail.com

Abstract

This report paper is based on the implementation and design of LPCXpresso 1769 CPU module interfacing with the RF module on the other board. Here the whole packet of data is sent to a different node, which has a similar RF module. The packet is made of sync field, source id, destination id and payload. The data except for sync field is scrambled, Linear Block coded and then sent to the other node. The sync field is verified by the LISA algorithm according to the given confidence level. Once the sync field is matched, the code word is checked if there is some error present in the received data. If there is some error, negative acknowledgment is sent. If there is no error, the remaining data of the packet is descrambled. After the descrambling, the destination id of the packet is matched with the source id of the node, if it matches then the data is shown as output.

1. Introduction

The main purpose of this lab is to implement the Linear Block Coding then display the payload using GPIO. The c program is written to perform the functionalities. The data transfer happens through the GPIO pin.

2. Methodology

The project is combination of both hardware and software and so this section demonstrates the methodology used for designing the circuit as well steps performed to write a code to achieve desired output.

2.1. Objectives and Technical Challenges

Objective of this project is to develop a circuit with LPC1769 module and RF board to demonstrate the transmission of data between them as well get the knowledge about working of Linear Block Coding.

There were some technical challenges faced while implementing the functionalities. One of the major technical challenges was to sync the bit rate between the nodes Ni and Nj. The other was to eliminate the noise to find the concerned packet and implement the LISA algorithm to check the confidence level from the synchronization field.

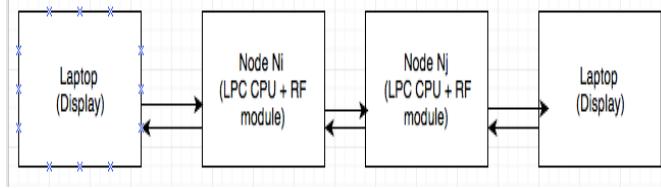
2.2. Problem Formulation and Design

To start the implementation of the given goal, the first thing needed was the embedded board formed by LPC 1769 module. The embedded board comprised of the LPC module, power regulator circuit and RJ 45 connector. The power regulator circuit is required for the regulation of voltage so that the voltage doesn't go beyond 5V. The voltage regulator circuit was made using the LM 7805 Regulator IC. Once the embedded board was made, the next stage was to build the RF board. The RF board comprises of 2 RJ 45 connectors, 1 FS 1000A transmitter and 1 MX-RM-5V receiver. The transmitter needed an external antenna to increase the transmission range.

The software portion is the implementation of the LISA algorithm, scrambler & descrambler, Linear Block Coding. In the LISA algorithm, the data is ideally sent in a packet. This packet has a 32-bit sync field, source id field, destination id field and the payload/data. The sync field is present to validate the data. In the sync field there is a pattern in binary consisting of numbers 50H to 5FH or A0H to A5H. All the data post the sync field is scrambled with a predefined scrambling order which is known by the sender and the receiver before hand. Once the data is scrambled, the scrambled data is multiplied with the generator matrix to form the code words. The code word are then transmitted. On the receiving side the node tries to match the pattern to check how much corruption is present in the packet. The number of patterns matched is called the confidence level. So, higher the confidence level higher the chances of data being less corrupt. The confidence level to be matched is taken as an input on the receiver side. If that level is matched then, the remaining data of the packet is multiplied with transpose matrix to obtain the syndrome word. If the syndrome word doesn't contain any error then the remaining data of the packet is descrambled. Once the data is descrambled, the destination id field of the packet is matched with source id of the node. If the ids match, then only the data is accepted and acknowledgment is sent. If the syndrome word has error then negative acknowledgment is sent to ask the sender to transmit the data again.

The whole process is implemented using a C language code. Here the program stores the incoming packet of data. It goes through the sync field, checks for

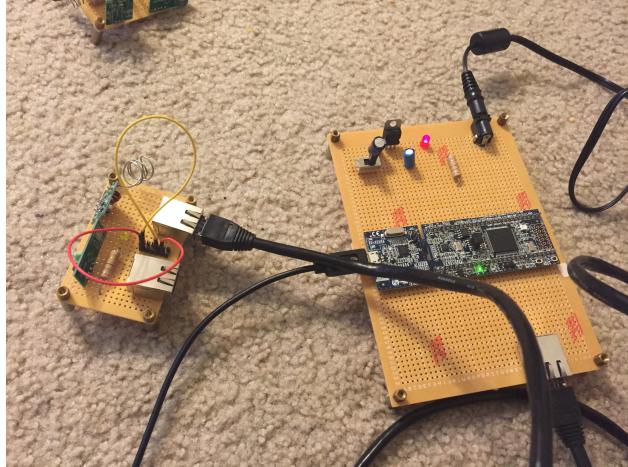
the matching patterns. The number of patterns to be matched depends upon the number of confidence level the user inputs. Once the pattern is matched, the syndrome word is formed to check if the error is present in the code words. The remaining portion of the packet is descrambled. After the descrambling, the destination id of the packet is matched with the source id of the node. If the ids match, then the data is displayed on the laptop screen. Once the receiver receives the data, the receiver sends the acknowledgment bit to the sender. Here the combination of Embedded board + RF board is called node Ni and the same combination on the other side is Nj. So, the communication is full duplex. As one node sends the data then receives the acknowledgment. The whole process is shown below in the block diagram.



3. Implementation

The implementation of the whole system is shown below:

3.1. Hardware Design



The circuit designs & schematic diagrams are as follows:

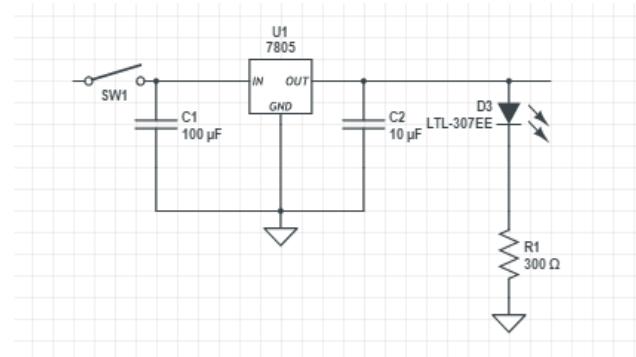


Fig1. Power Regulator Circuit

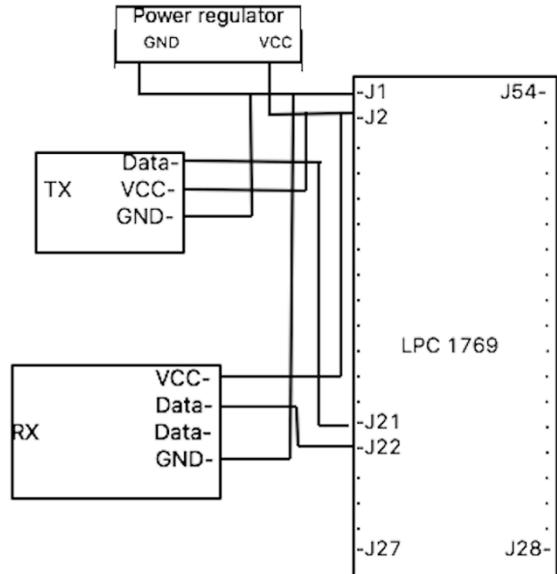


Fig2. Schematic Design of the whole circuit

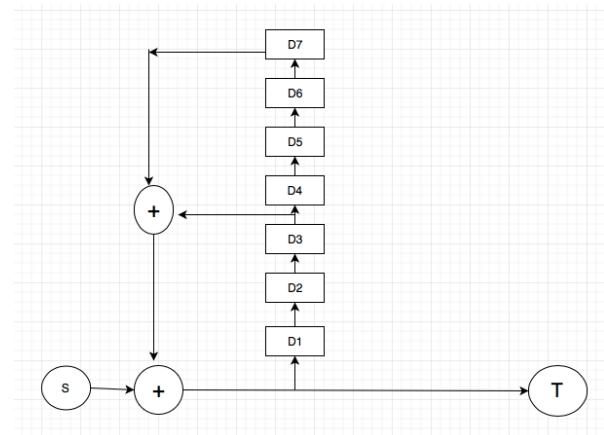


Fig3. Scrambler Circuit (7th Order)

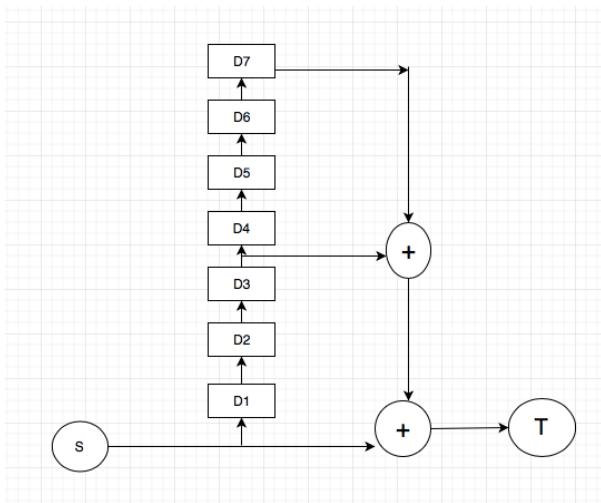


Fig4. Descrambler Circuit (7th Order)

Hardware Materials

- LPCXpresso ARM Cortex 1769 CPU module
- FS 1000A
- MX-RM-5V
- LM 7805

Bill of Materials

Item	Description	Quantity
Board	3" x 2" Prototype board	1
Board	6" x 8" Prototype board	1
Connector	RJ 45 J1 Connector	3
Cable	RJ 45	2
Capacitor	100 uF	1
Capacitor	10 uF	1
Resistor	300 Ohm	2
LED	Red & Yellow (5V)	2
Switch		1
Power Adopter	5 V	1
Power Adapter Connector	Cable Connector	1
Standoffs	Board Standoffs	8
Pin heads	6 pin dual Pin head	1

GPIO Design

Pin	Description
Pin 1	VCC
Pin 2	Data In
Pin 6	Data Out
Pin 7	GND

3.2. Software Design

Flowcharts:

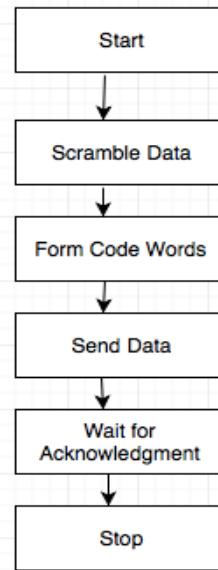


Fig 3. Flowchart for sender Node

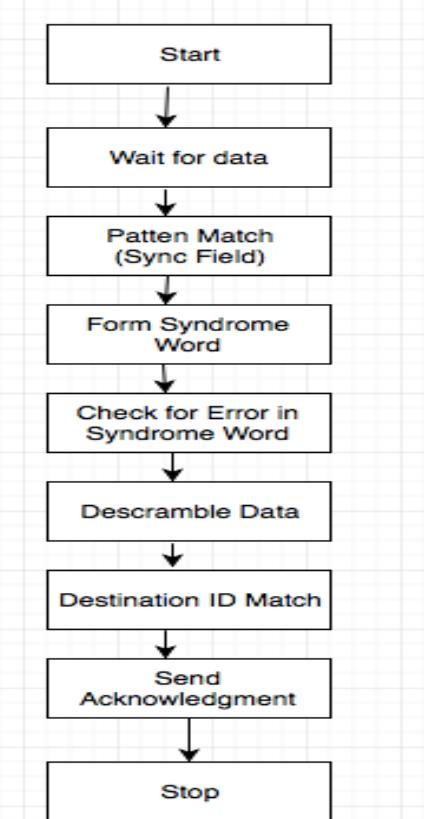


Fig 4. Flowchart for Receiver Node

Step by Step Algorithm

1. Starting the module and deploying the code on LPC 1769
2. Scramble Data
3. Form the code words
4. Send the Data to Node j from Node i
5. Waiting for the acknowledgment
6. Received Acknowledgment
7. Waiting to receive data from Node j
8. Performing pattern matching in the sync field
9. Calculating the confidence level and comparing to the pre defined confidence level
10. Form the syndrome word
11. Check error in the syndrome word
12. Descrambling Data
13. Matching Destination id with the source id
14. Sending acknowledgment once the ids are matched
15. Extracting Payload
16. Displaying the Payload on the screen

4. Testing and Verification

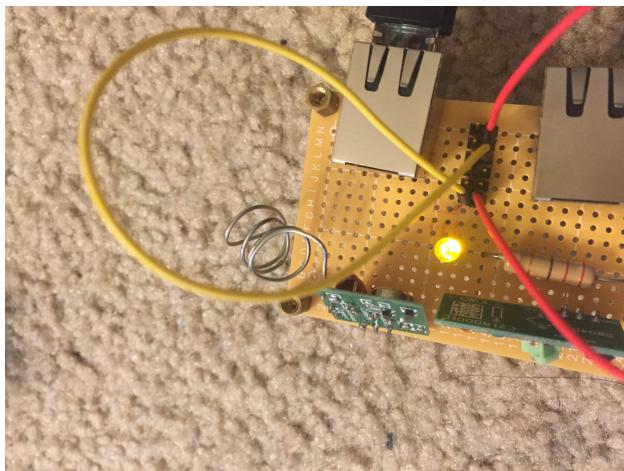
Below are the test cases used for testing the implementation:

Test Case 1: Data sending to the RF board

Input: Data from the LPC Module

Expected Output: LED set on RF module

Output: LED set on RF module



Test Case 2: Data sending from Node i to Node j

Input: Data to be sent to Node j

Expected Output: Acknowledgment Received

Output: Acknowledgment Received

```
LPC1769_GPIO Debug [C/C++ (NXP Semiconductors) MCU Application]
1
Acknowledgment ReceivedEnter the confidence level:
7
Enter the order for scrambling:
7
Enter 1 for scrambling and 2 for descrambling:
1
Enter 1 for send and 2 for receive.
1
Acknowledgment ReceivedEnter the confidence level:

Writable Smart Insert 1010:1
```

Test Case 3: Data Receiving and implementing LISA

Input: Data from the from the Node j

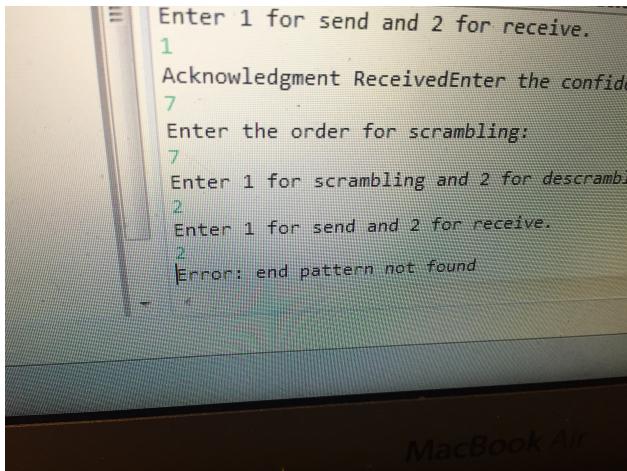
Expected Output: Sync pattern match & displaying payload

Output: Sync pattern match & displaying payload

```
Console Problems Memory Instruction Trace SWO Trace
LPC1769_GPIO Debug [C/C++ (NXP Semiconductors) MCU Application] LPC1769_GPIO
Enter 1 for send and 2 for receive.
2
End pattern matched
endbit is 271Sync matched with the last pattern being 54
Destination start bit is 139
payload start value is=151
Destination id of the incoming packet matched with the source id.
PARTH 0995
Enter the confidence level:
7

Writable Smart Insert 1010:1
```

Test Case 4: Corrupted Data from the Sender
Input: Corrupted Data from the Node j
Expected Output: Confidence level not matched, data not to be accepted
Output: Confidence level not matched, data not to be accepted



[4] FS 1000A Transmitter and MX-RM-5V Receiver Datasheet
<http://wiki.jmoon.co/sensors/rfrxtx/>

5. Conclusion

The conclusion of performing this lab was to get the complete picture of LPC 1769 module, RF module, LM 7805 based Power regulator circuit, the LISA algorithm, scrambler and descrambler & Linear block coding. The packet was transferred from one RF board to the other and receiving the same from the other RF board. Once the data was received, the LISA algorithm was performed to extract the payload. By doing this, goals were achieved and the above aim was concluded.

6. Acknowledgement

My Personal Acknowledgement to Professor Harry Li. for providing the proper guidance and the fundamental information required for the project completion.

7. References

[1] Harry Li. "CMPE 245 Lecture Notes", Computer Engineering Department, College of Engineering, San Jose State University

[2] LM 7805 Datasheet
<https://www.fairchildsemi.com/datasheets/LM/LM7805.pdf>

[3] LPCXpresso 1769 Datasheet
https://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf