AN1201A.06
Application Note

How to make a point to multi-point communication with the XE1201A

Author : Gael Coron
For further information, please contact

XEMICS SA
E mail: info@xemics.com
Web: www.xemics.com

Table of Contents

INTRODUCTION

This application note describes a point to point and point to multi-point communication protocol, which is used in the XE1201A - CTD (Complete Transceiver Demonstrator). The source code of this demonstrator is written in C and can be downloaded, free of charge, from the XEMICS web site: http://www.xemics.com

The objective of this demonstrator is to provide a hardware solution using the XM1201A (RF module) and software running on the XE88LC05.

Most of applications are based on small networks such as clusters or piconets. It means being able to communicate from one device to n devices. To achieve this type of transmission, different approaches can be used. One of them is the Time Division Multiple Access (TDMA) protocol.

GENERAL DESCRIPTION

This transmission protocol example shows a communication between 1 master and a maximum of 64 slaves maximum. To have high flexibility while using the same hardware, different modes can be implemented by software:

- Point to point or multipoint communication.
- Automatic or manual
- Master or slave

This example is based on the Music Box: Each Music Box contains two different boards:

- RF module : XM1201A
- System application board: based on the microcontroller: XE88LC05

The XM1201A is a complete RF module based on XEMICS 300-500MHz transceiver. Information on the XM1201A can be found in the application note AN1201A.03. The XE88LC05 is an 8bit microcontroller with on-chip AD/DA converters.

KEY APPLICATIONS

The XM1201A gluelessly interfaces to the XE88LC05. Together with this TDMA communication protocol, the solution is ideally suited for application such as:

- Home automation
- Wireless sensing
- Remote control
- Security systems
- Process and building control

SYSTEM BOARD

The demonstrator consists of two boards (Figure 1): the system board and the RF module (XM1201A). Both board are interconnected via 10 pins connector. Please note that this document outlines the features and operating modes of the system board. For information on the XM1201A, please refer to AN1201A.03

The system board can be configured as “master or slave”. This selection is made via a push-button and so two different modes.

The source code was written to allows for different types of communication:
“Point to Point”, this example uses two boards one master, one slave.

“Point to multi-point”, this example allows to use several slaves with one master.

The demonstrator can also operate in manual or automatic mode.

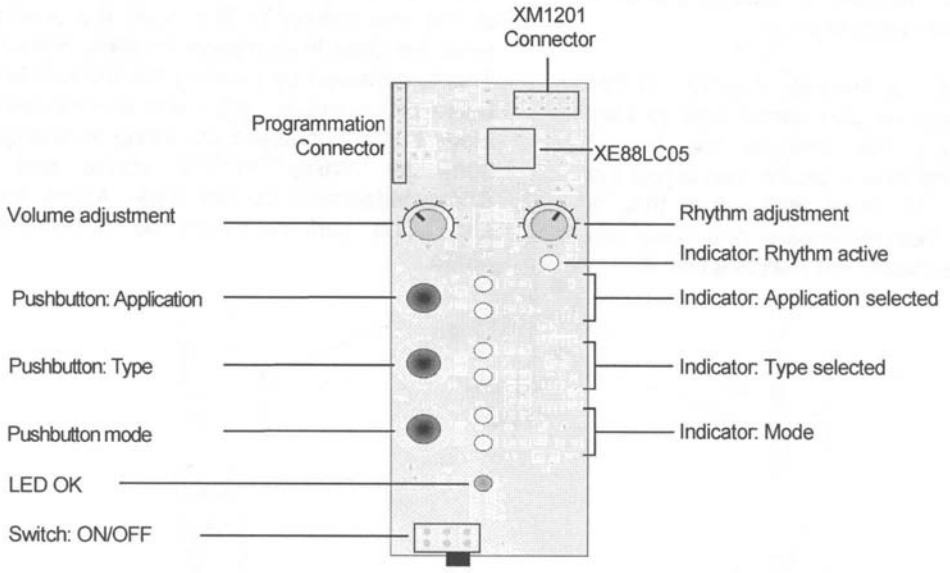

Figure 1: Application board

Name	Description
Volume adjustment	Allows to adjust the volume of the loud speaker
Rhythm adjustment	Allows managing the music rhythm. This adjustment is enabled only if the board is the master of the network.
Pushbutton: Application	Selects the application: Point to point / Multi-point
Indicator: Application	Indicates which application is selected
Pushbutton: Type	Selects which type of board: Master / Slave (Only one master can be set-up)
Indicator: Type	Indicate the type of the board (Master of Slaves)
Pushbutton: Mode	Select the mode: Automatic / Manual
Indicator: Mode	Indicate which mode has been selected
LED OK	The led is: Turned on at power up Turned off if the board has been configured as a master or if it has been configured has slaves and recognized by the master. Blinking, when the board is playing music (switch on)

POINT TO POINT COMMUNICATION

The first objective of this example is to show an RF link between two boards, a master and a slave (Figure 2). If a third board is switched on, it will be recognized by the master but not included in the communication.

The point to point communication takes place in a ping-pong fashion where the master sends data to the slave, upon receipt of the data, the slave sends an acknowledgement back to the master. This cycle is reiterated as long as the communication is active. To show that a point to point communication is taking place each demonstrator unit will play a music tone when it receives data from another unit.

"Automatic Mode" is the default mode. In this mode the master decides which tone to play and when, only the user, via the rhythm adjustment on the master board, can adjust the tone duration. In order to insure that a communication between master and slave can be activated the master first makes a test. A

"Test here" command is sent by the master. The master then awaits an acknowledgement form the slave. Once the acknowledgement received, the point to point communication can begin. Each demonstration unit plays a music tone when data is being exchanged: the master when it sends data to the slave and the slave when it sends the acknowledge to the master. If the speaker volume is switch off, a red LED is switched on to inform the user that a tone is played.

"Manual Mode". By pressing the "Pushbutton Mode" on the master, the application switches into manual mode. A green LED on the master board and on the slave board, informs the user of this new setting. In this mode the user can send the data from master to slave manually. This is achieved by pushing the Manual Mode button on the master unit. Each time the Manual Mode button is pressed one string of data goes from the master to the slaves and its acknowledgement comes back. Music tones and LEDs perform similarly as in Automatic Mode.

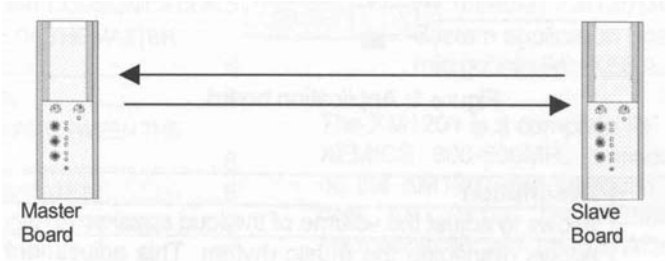

Figure 2: Point to point communication

POINT TO MULTI-POINT COMMUNICATION

The point to multi-point example (Figure 3) allows to achieve a transmission between one master and n slaves. Each slave plays a tone when being addressed by the master who manages the communication. On power up the demonstrator units are all in slave mode. To realize a point to multipoint communication, one unit must be in Master mode and all units “Multipoint” mode.

The “Manual mode” is accessible via the master board. The user needs to push on the button to send the message (the tone to play) to the next slave.

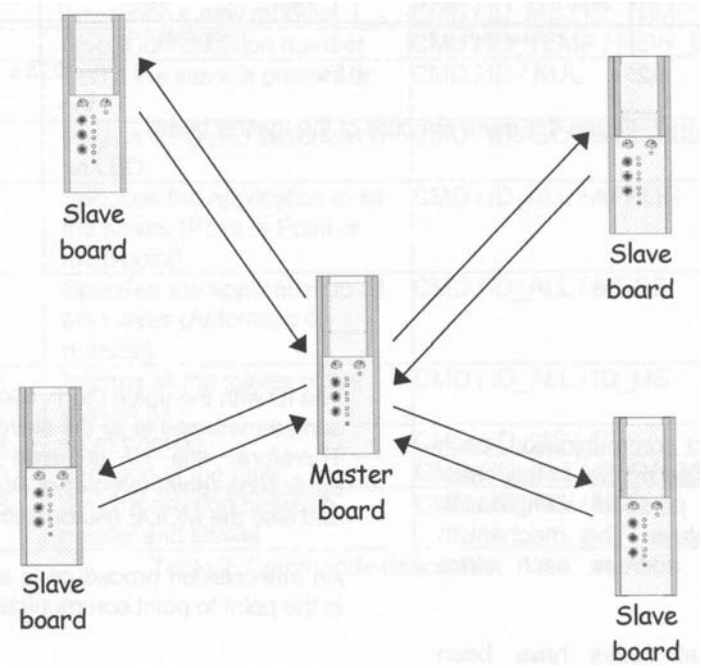

Figure 3: Point to multipoint communication

SOFTWARE STRUCTURE

TIME BASE

The communication protocol is based on the reference time. Each 0.25 second, the master sends to all the slaves a message with information about the application (type of application, the mode, and the rhythm). Between two transmissions, the master switches in receiver mode to detect the presence of new slaves. To synchronize the new boards, a dedicated frame is sent every second from the master. The “Time State” labels are used to determine the action to do (i.e. Send synchronization frame or listen for new slaves).The Figure 4 shows this communication structure.

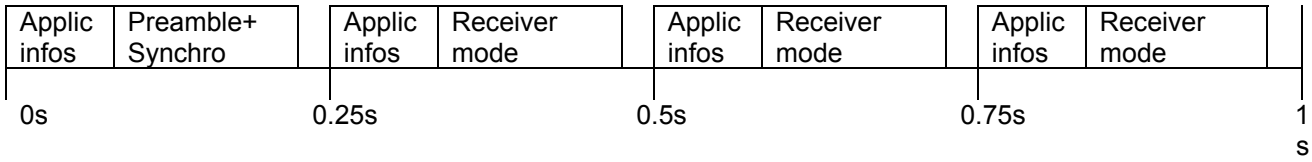

Figure 4: Frame strucure of the master board

SLAVES IDENTIFICATION

For a point to multi-point communication, each slave needs to be identified by the. In this case, the master assigns a personal identification number (ID) to each slave. This mechanism allows the master to address each slave individually.

However, some special codes have been defined, for example a command that allows addressing all the slaves at the same time.
The ID with the value 0xFF, allows the master to send a message to all the slaves in the network. Therefore, this ID is used to change the application (point to point or Point to multi point) and also the MODE (Automatic or manual).

An anti-collision procedure is also implemented in the point to point communication protocol.

FRAME STRUCTURE AND COMMANDS LIST

All RF messages have a predefined length. It allows to have a simple communication procedure for transmission and reception. Each frame is constructed as described in Figure 5:

Each frame is 72 bits wide (see Table 1).

To cover all the needs, 10 commands have been defined (see Table 2 and 3).

Preamble	Start word	CMD	ID	CPL	CRC

Figure 5: Frame structure

Name	Description	Length
Preamble	Synchronization frame	32 bits
Start word	Start communication	8 bits
CMD	Command	8 bits
ID	Slave Identification number	8 bits
CPL	Complement of command	8 bits
CRC	CRC Code, no corrective based on CMD, ID and CPL	8 bits

Table 1: Frame structure description

CMD	CMD Code	Description	Frame structure	Answer
New_Module	0x01	Announce a new module	CMD / ID_MS / ID_TEMP	Attribute_ID
Attribute_ID	0x02	Give a identification number	CMD / ID_TEMP / NEW_ID	ACK
Test_Here	0x03	Test if the slave is present or not	CMD / ID / NUL	ACK
Play_Song	0x05	Plays a song and switches on an LED	CMD / ID / SONG NUMBER	ACK
Application	0x06	Specifies the application to all the slaves (Point to Point or multi-point).	CMD / ID_ALL / APPLIC	-
Auto_Manual	0x07	Specifies the application to all the slaves (Automatic or manual).	CMD / ID_ALL / MODE	-
ID_Master	0x08	Informs all the slaves of the master ID	CMD / ID_ALL / ID_MS	-
ACK	0x09	Confirmation	CMD / ID_MS / ID	-
Song_Duration	0x0A	Duration of the tone	CMD / ID_ALL / DURATION	-
Synchro	0x0B	Synchronization between master and slaves.	CMD / ID_ALL / NUL	-

Table 2: Commande description

Code name	Code	Description
ID_MS	0x00	Master identification number
ID_TEMP	0x80 < ID_TEMP < 0xFF	Random time used by a new module
NEW_ID	0x00 < NEW_ID < 0x80	Final identification number for a slave given by the master
ID	0x00 < ID < 0x80	Identification of a dedicated slave
NUL	0x00	Complement of command
SONG_NUMBER	0x00 = No tone 0x01 – 0x08	Tone number to generate
ID_ALL	0xFF	Identification for all the slaves
APLLIC	0x01 = point to point 0x04 = Multi-point	Application definition
MODE	0x03 = Automatic 0x06 = Manual	Mode definition
DURATION	0x02 – 0x10	Time duration of the tone, the time base is 0.25s

Table 3: Hexadecimal code allocation

SOFT ROUTINE EXAMPLES

•Slave detection and communication

The following schematic (Figure 6) explains how to achieve an RF communication between a master and a slave. This example consists of sending a command (Play song). Before the transmission can begins the master tests if the slave is present.

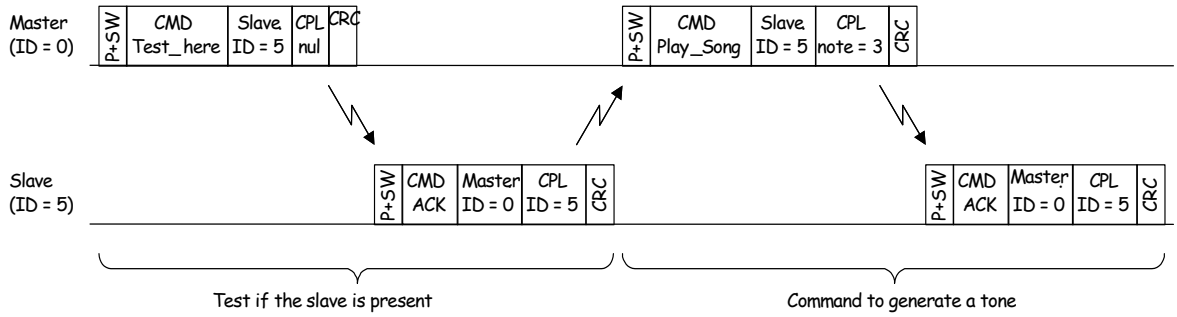

Figure 6: RF communication between the master and a slave

NEW SLAVE IDENTIFICATION AND ANTI-COLLISION

When a new slave comes into the network, it will try to communicate with the master after a random time. This random time is generated by a random counter. Therefore, the new module will send a message to the master by using an ID equal to 128 plus the value of the random counter (ID_TEMP). The master confirms the transmission by sending to the actual slave (ID_TEMP) and a personal ID (ID_NEW). This personal ID is number inferior to 128.

The following schematic (Figure 7) shows the protocol for the identification of a new slave.

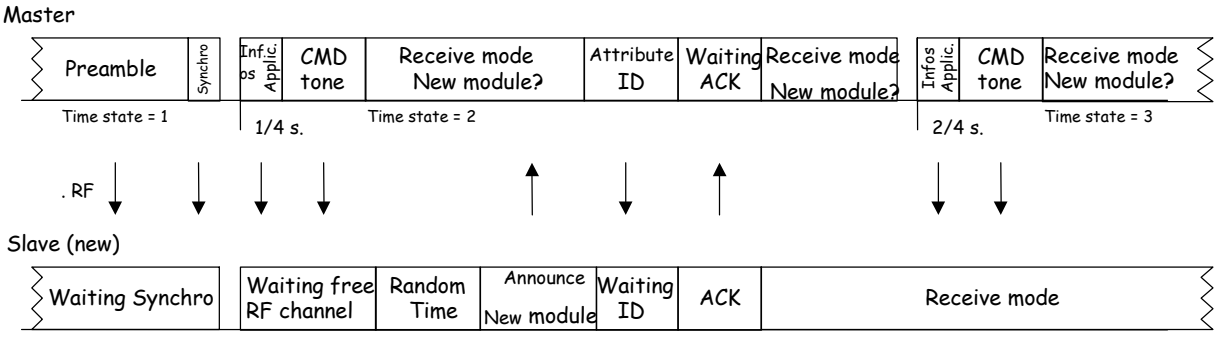

Figure 7: New slave identification

Communication initialization routine

Flow chart

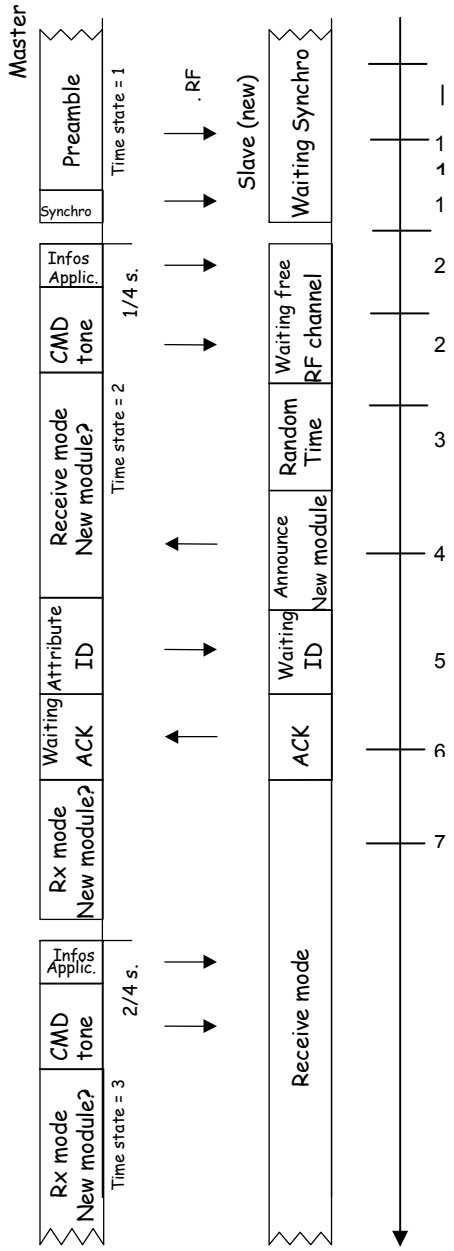

Figure 8: Identification flow chart

The Figure 8 shows:

1. At power up, the slave boards switches into receiver mode and awaits the Synchro frame from the master. In this way, the slaves are synchronized with the master.
2. After the synchronization frame, the master sends to all the slaves the information regarding the application (Mode, Application, and Tone duration). After, if applicable, the master send a message “Play a song” to one slave. In the mean time, a new slave is waiting until the master finishes the transmission.
3. The master switches into receiver mode and listens if there is a new slave in the network. The slave waits a random period of time and assigns itself an address, which is based on its random counter.
4. The slave board sends a message to the master with its temporary address.
5. The master board answers by giving a permanent address.
6. The slave sends an acknowledgment to confirm the ID reception. The master can now reserve this address for the new slave.
7. At this point, the slave is identified and can participate in the point or point or multi-point communication.

SOURCE CODE

FILES

File name	Description
Conv_adda.c / Conv_adda.h	Procedure for the analog chain
Demon.c	Main file
Demon.h	Constants and types definition for the main file
InitFVe.h	Constants and types definition for the analog chain
InitReg.h	Constants and registers definition for the microcontroller XE88LC05
IO_def.h	Inputs and outputs definition
IRQHandler.c	Interrupt procedures
Proc.c / Proc.h	General procedures
RF_commands_lib.c / RF_commands_lib.h	Data processing procedures for RF
RF_module_Lib.c / RF_module_Lib.h	XE1201A programmation procedures

Table 4: Files list name

XE1201A PROGRAMMATION PROCEDURES

Procedure name	Description
InitRFmodule	Initialization and configuration of the RF module
WregABC	Configuration of XE1201A registers (A, B and C)
OpenRF	Switch on the RF module
CloseRF	Switch off the RF module
WRF_Synchro	Transmission of the Preamble and synchronization frame
RRF_Synchro	Reception of the Preamble and synchronization frame
WRF_CMD	Transmission of one command
RRF	Reception of one command

Table 5: XE1201A procedure list name

PROCEDURE FOR THE ANALOG CHAIN

Procedure name	Description
Init_song	Analog chain initialization
Play_song	Tone generation
Get_song_Duration	Read the value of the “tempo” trimmer
GET_Song_Frequency	Frequency measurement for tone generation

Table 6: Analog chain procedure list name

DATA PROCESSING PROCEDURES FOR RF

Procedure name	Description
Decode_CMD	Decode the received message and execute the command
TEST_n_Play	Give the order to generate a tone to the next available slave
FIND_Present	Find the next module
WAIT_Synchro	Synchronization of slave with the master

Table 7: RF procedure list name

EXIBIT: SCHEMATIC