Lab 3

Protocol Layering with the RFM12B-S2 Radio Module

The aim of this third lab is to incorporate the existing functionality that you have already developed into a basic interrupt-driven layered protocol, using the RFM12B-S2 wireless transceiver with the Il Matto board.
Schedule

Preparation time : n/a
Lab time : 3 hours

Items provided

Tools : n/a
Components : RFM12B-S2 Radio Module (1 each)
Equipment : Oscilloscope, Logic Analyser
Software : n/a

Items to bring

Essentials. A full list is available on the Laboratory website at https://secure.ecs.soton.ac.uk/notes/ellabs/databook/essentials/

Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Authors</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 30, 2017</td>
<td>Domenico Balsamo and Geoff Merrett</td>
<td>Revised for 2017/18</td>
</tr>
<tr>
<td>October 22, 2016</td>
<td>Domenico Balsamo and Geoff Merrett</td>
<td>Revised for 2016/17</td>
</tr>
<tr>
<td>October 26, 2015</td>
<td>Domenico Balsamo and Geoff Merrett</td>
<td>New lab for 2015/16</td>
</tr>
</tbody>
</table>

© Electronics and Computer Science, University of Southampton
1 Aims, Learning Outcomes and Outline

These lab exercises will provide an introduction to computer networks and wireless communication using the Il Matto development board. In particular, you will learn how to use the RFM12B-S2 wireless transceiver (transceiver = receiver + transmitter) to enable communication between two Il Matto boards, and the basics of how to implement a protocol stack.

In this lab, you will explore part of a basic interrupt-driven layered model using the RFM12B-S2 wireless transceiver with the Il Matto board. An interrupt-driven layered model is commonly used when a layer needs to report changes or events in its state to the above layer and such changes often induce new assignments or actions at the upper layer. As known from Lab 2, the RFM12B library already implements the PHYsical layer and some functionalities related to the Media Access Control (MAC) sublayer of the Data Link Layer (DLL) layer (e.g. Carrier Sense Multiple Access, CSMA, protocols). However, these functionalities have not been organized considering a proper layered implementation. Your first objective is to encapsulate these functionalities into separate layers (a PHYsical layer and MAC sublayer).

A second objective (this is additional work for those that finish the first objective) is to add a basic service interface between the Logical Link Control (LLC) sublayer and Media Access Control (MAC) sublayer, defining and complying with primitives and service definitions between layers.

2 Preparation
None required.

3 Laboratory work

You should work in pairs for this lab exercise. Each pair should ensure that they have at least one working Il Matto between them, and that at least one person is familiar with the Il Matto and its programming.

Your first objective is to encapsulate the functionality provided by the RFM12B library into separate layers (a PHYsical layer and MAC sublayer).

Your second objective (additional work for those that finish the first objective) is to add a basic service interface between the Logical Link Control (LLC) sublayer and Media Access Control (MAC) sublayer, defining and complying with primitives and service definitions between layers.

3.1 Protocol Layering

As presented in the ELEC3222 lectures, protocol layering (or protocol stack) is used to divide networking functionality into more manageable chunks, also permitting code reuse. Each layer (the service provider) provides services to the upper level (service user) through a service interface while communication occurs virtually between peer entities. Services are provided to the layer above as primitives (i.e. functions) through service access points (SAPs).
There are four main types of service primitive (see Figure 1):

- **Request**: the service user (upper layer) issues this primitive to the service provider (layer below) to perform a task, e.g. transmitting data;
- **Confirm**: the response to a previous request that has come back, e.g. data successfully transmitted);
- **Indication**: the service provider issues this primitive to the service user to indicate that an event has taken place in the peer service of the remote protocol stack;
- **Response**: this primitive is issued by the service user in response to an event from the indication primitive received;

![Fig. 1: Service primitives](image)

### 3.2 Lab Tasks

As known from Lab 1 and Lab 2, the RFM12B library already implements the PHYsical layer and some functionalities related to the Media Access Control (MAC) sublayer of the Data Link Layer (DLL) layer (e.g. Carrier Sense Multiple Access, CSMA, protocols). However, these functionalities have not been organized considering a proper layered implementation.

In this lab, your first objective is to encapsulate these functionalities into separate layers (a PHYsical layer and MAC sublayer), see Figure 2.

![Fig 2: PHYsical layer and MAC sublayer](image)
Specifically, you already explored link access and broadcasting data across a shared channel using the RFM12B-S2 module, by implementing CSMA protocols. However, this has been done without ‘breaking up’ that RFM12B-S2 library driver and putting these functionalities into the right layers (PHYsical and MAC layers), and using the right service primitives.

In this lab, you have to encapsulate the two CSMA functionalities in two different layers:

1. The part of the code that controls the hardware to take and return a measurement if the channel is being used or not (or an average RSSI value) should to be located in the PHYsical layer;

2. The part of the code that controls the flow of backing off and resending frames should be located in the MAC layer.

These two parts of the code in the current version of the RFM12B-S2 library are merged within the same function rfm12_tick() that is invoked in the main(), but this implementation doesn’t follow the desired layered structure.

Test the code to make sure it’s working. How do you know that it’s working and actually doing (exactly) what it’s supposed to?

### 3.3 Additional Work

If you have enough time, you can also attempt to add the following functionalities:

1. Implement a basic service interface between the Logical Link Control (LLC) sublayer and Media access control (MAC) sublayer (see Figure 3), and defining and complying with primitives and service definitions between layers.

![Diagram](image.png)

**Fig 3:** The DLL, made up from the LLC and MAC sublayers

A sequence diagram illustrating the use of the service primitives introduced before is shown in Figure 4. Implement the service interface described below:

2. For CSMA, the `confirm` primitive will indicate that a frame of data associated to a previous `request` has (or has not) been successfully transmitted. Each service primitive has a number parameters associated with it.

3. For example, the transmit `request`, that we named `TRAN_CSMA.request`, should include the destination address and a Service Data Unit (SDU), while the `TRAN_CSMA.confirm` should include a parameter that specifies the success or failure.
4. If \textit{confirm} primitive indicates success, this means that the data at the layer below has been successfully transmitted through the channel (but we do not know if the data has been received by the peer entity) while a failure should indicate why the transmission failed (e.g. excessive collisions, carrier detected, etc.).

5. The \textit{TRAN_CSMA.indication} primitive issued by the MAC layer informs the LLC layer that a frame has been received.

![Service primitives for CSMA](image)

**Fig. 4 Service primitives for CSMA.**

### 4 Resources

You can find more information about RFM12B-S2 on-line:

- [https://www.sparkfun.com/products/12031](https://www.sparkfun.com/products/12031)

It is very useful to use/read the following documents:

- RFM12B-S2 Datasheet:

- Programming guide:
  - [https://www.sparkfun.com/datasheets/Wireless/General/RF12B_code.pdf](https://www.sparkfun.com/datasheets/Wireless/General/RF12B_code.pdf)

- RFM12B IC Datasheet:
  - [https://www.sparkfun.com/datasheets/Wireless/General/RF12B-IC.pdf](https://www.sparkfun.com/datasheets/Wireless/General/RF12B-IC.pdf)