Alexandria University
Faculty of Engineering
Computer and Systems Engineering Department

Embedded Systems
Assigned: Sunday 24th Apr, 2016
Due: Wednesday 27th Apr, 2016

Arduino LAB

Assignment 5

Wireless communication:

The nRF24L01 is a single chip 2.4GHz transceiver, designed for ultra low power wireless applications.
The nRF24L01 is designed for operation in the world wide ISM frequency band at **2.400 - 2.4835GHz**.
An MCU (microcontroller) and very few external passive components are needed to design a radio system with the nRF24L01.

The nRF24L01 is configured and operated through a Serial Peripheral Interface (SPI.) Through this interface the register map is available. The register map contains all configuration registers in the nRF24L01 and is accessible in all operation modes of the chip.

For more details about the operation of the nRF24L01 transceiver module please refer to:
http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01

Arduino Connection

Signal	RF Module PIN	Arduino pin for RF24 Library	MEGA2560 pin RF24 Library
GND	1	GND *	GND *
VCC	2	3.3V *	3.3V *
CE	3	9	9
CSN	4	10	53
SCK	5	13	52
MOSI	6	11	51
MISO	7	12	50
IRQ	8	-	-

Arduino Library:
Please refer to this link to understand how the library for the nRF24 module works and the main functions:-
https://arduino-info.wikispaces.com/Nrf24L01-2.4GHz-HowTo
Below are screenshots from this link highlighting the most important functions.
The link for the library on github:- https://github.com/tmrh20/RF24
Additional tutorial similar to the assignment problem:
https://arduino-info.wikispaces.com/Nrf24L01-2.4GHz-ExampleSketches#bm1

CREATE AND START UP A RADIO (For either transmit or receive):

RF24 (uint8_t _cepin, uint8_t _cspin) Create a radio and set the Arduino pins to be used for CE and CS)

EXAMPLE: (Create an instance of a radio, specifying the CE and CS pins.)
RF24 **myRadio** (7,8); *"myRadio" is the identifier you will use in the following examples*

NOTE: *The following pins are fixed and unchangeable. AND Vcc (supply voltage MUST be 3.3V)*
SCK to Arduino pin 13
MOSI to Arduino pin 11
MISO to Arduino pin 12

NOTE: *In all following commands, you must use the same identifying name you used in the RF24 statement that created the radio. ("**myRadio**" in the example above) You will use that identifier followed by (dot) followed by the method (command), followed by parameters. This is "object oriented programming". The OBJECT in this case is the Radio, and there are many METHODS that can operate on the Object. A METHOD is much like a Function or a Subroutine.*

EXAMPLE: ***myRadio.begin();*** Start up the actual radio module with the "begin" method

NOTE: "PIPES" : This is often confusing. nRF24L01 uses "pipes" that connect from transmitter to receiver. Pipes have an address you need to set. The Transmitter pipe must have the same address as the Receiver pipe. Later it's possible to use multiple "pipes" at once

EXAMPLE OF RECEIVING DATA:

myRadio.openReadingPipe (1, const uint8_t *address) Pipe number (usually 1), pipe address (which is usually 5 bytes in an array structure).

EXAMPLE:
byte addresses[][6] = {"1Node"}; Create address for 1 pipe.
myRadio.openReadingPipe(1, addresses[0]); *Use the first entry in array 'addresses' (Only 1 right now)*

myRadio.startListening (); *Turn on the receiver and listen for received data. You MUST have opened a reading pipe FIRST.*

if(***myRadio.available()*** Check for available incoming data from transmitter
{
while **(myRadio.available())** *While there is data ready*
{
myRadio.read(*&myData, sizeof(myData)) ;* Get the data payload (You must have defined that already!)
}
myRadio.stopListening(); *stop listening*

EXAMPLE OF TRANSMITTING DATA:
byte addresses[][6] = {"1Node"}; Create address for 1 pipe.
myRadio.openWritingPipe(1, addresses[0]); *Use the first entry in array 'addresses' (Only 1 right now)*

myRadio.write(&myData, sizeof(myData) *)*

SET SOME OTHER OPERATING OPTIONS AND PARAMETERS:
NOTE: Many of these methods have default values you can usually use.

radio.printDetails();
Prints out a LOT of debugging information.

radio.setDataRate(RF24_250KBPS);
speed RF24_250KBPS for 250kbs, RF24_1MBPS for 1Mbps, or RF24_2MBPS for 2Mbps. 250K Bits per second gives longest range.

radio.setPALevel(RF24_PA_MAX);
Set Power Amplifier (PA) level to one of four levels: RF24_PA_MIN, RF24_PA_LOW, RF24_PA_HIGH and RF24_PA_MAX
The power levels correspond to the following output levels respectively: NRF24L01: -18dBm, -12dBm,-6dBM, and 0dBm
0 dBm is equal to 1 milliwatt. Each 3 dB is a 2:1 ratio. 6 dB is 4:1 10 dB is 10:1 So -18 dBm is 0.0000158489 watts !
To calculate all this dBm Stuff (Which us old Radio Engineers love) See THIS..
The "High Power" nRF24L01 modules like THIS have a gain of 100, so their output is +20 dBm (100 milliwatts)

radio.setChannel(108);

Which RF channel to communicate on, 0-124 Can operate on frequencies from 2.400GHz to 2.524GHz. Programming resolution of channel frequency is 1Mhz
This is the same unlicensed band WiFi operates in (WiFi uses 2.400 to 2.500 gHz). Usually frequencies above channel 100 are best.

Requirements:-

You are required to use RF communication to control the operation of an LED using an ON-OFF switch.
Two arduinos are used (UNO & Mega), both are connected to a different nRF24L01 module.
One of them is acting as a transmitter connected to the ON-OFF switch, the other is connected to the LED and acting as receiver.
Once the switch is ON, the LED should become ON. Once the switch is OFF, the LED should become OFF.
It is required to make proper connections and develop a working code for both the transmitter and the receiver.

Note: In order to connect the nRF24 with the Arduino, you will need female-to-male jumpers since the module pins cannot be used on breadboards.

Deliverables

- Submit a small video demonstrating the experiment given that your video will clearly show the following:-
  - Clear view of the connections of both Arduinos (Mega & UNO) with the transcievers, on-off switch and the LED.
  - Clear view of the part of code where the **transmit/receive frequencies** are set.
  - The effect of pressing the switch on the LED in the same frame for 3 consecutive times (ON – OFF – ON).
- Send your reports to: omar.salaheldine@gmail.com with the subject: Embedded_LAB_5. Make sure to include your full names and IDs in the mail body.

Good Luck