# 11 WIRELESS COMMUNICATION - BLUETOOTH and ZIGBEE – Code Snips

**Note: These Code Snips are taken straight from the book chapter; i.e. the “Program Examples”. In some cases therefore they are not complete programs.**

/\* Program Example 11.1: Bluetooth serial test data program

Data is transferred from mbed to PC via Bluetooth. \*/

#include "mbed.h"

Serial rn41(p9,p10); //name the serial port rn41

BusOut led(LED4,LED3,LED2,LED1);

int main() {

rn41.baud(115200); // set baud for RN41

while (1) {

for (char x=0x30;x<=0x39;x++){ // ASCII numerical characters 0-9

rn41.putc(x); // send test char data on serial to RN41

led = x & 0x0F; // set LEDs to count in binary

wait(0.5);

}

}

}

Program Example 11.1: Bluetooth serial test, mbed to PC

/\* Program Example 11.2: Bluetooth serial test data program

Data is transferred bidirectionally between mbed and PC via Bluetooth.

\*/

#include "mbed.h"

Serial rn41(p9,p10); //name the serial port rn41

BusOut led(LED4,LED3,LED2,LED1);

int main() {

rn41.baud(115200); // setup baud rate

rn41.printf("Serial data test: outputs received data to LEDs\n\r");

while (1) {

if (rn41.readable()) { // if data available

char x=rn41.getc(); // get data

led=x; // output LSByte to LEDs

}

}

}

Program Example 11.2: Bluetooth bidirectional data test

/\* Program Example 11.3: Paired Bluetooth master program

\*/

#include "mbed.h"

Serial rn41(p9,p10); //Name and define the serial link to the RN-41

BusOut led(LED4,LED3,LED2,LED1);

DigitalIn Din(p26); // digital switch input on pin 26

char x;

void initialise\_connection(void);

int main() {

rn41.baud(115200);

wait(1.0); //can only enter Command mode 500ms after power-up

initialise\_connection();

while (1) {

if (Din==1) // if digital input switched high

x=0x0F; // override count, with 0x0F

else {

x++; // else increment and

if (x>0x0F) // output count value

x=0;

}

rn41.putc(x); // send char data on serial

led = x; // set LEDs to count in binary

wait(0.5);

}

}

void initialise\_connection() {

rn41.putc('$'); // Enter command mode

rn41.putc('$'); //

rn41.putc('$'); //

wait(1.2);

rn41.putc('S'); //enter Master mode

rn41.putc('M');

rn41.putc(',');

rn41.putc('3'); //"Auto-connect" Master mode

rn41.putc(0x0D); //CR

wait(0.1);

rn41.putc('C'); // Causes RN-41 to attempt connection

rn41.putc(','); // Send MAC address of target slave device

rn41.putc('0'); //

rn41.putc('0'); //

rn41.putc('0'); //

rn41.putc('6'); //

rn41.putc('6'); //

rn41.putc('6'); //

rn41.putc('7'); //

rn41.putc('3'); //

rn41.putc('D'); //

rn41.putc('2'); //

rn41.putc('D'); //

rn41.putc('4'); //

rn41.putc(0x0D);

wait(1.0);

rn41.putc('-'); // Exit command mode

rn41.putc('-'); //

rn41.putc('-'); //

rn41.putc(0x0D); //

wait(0.5);

}

Program Example 11.3: Paired Bluetooth Master program

/\* Program Example 11.4: Bluetooth paired slave program

\*/

#include "mbed.h"

Serial rn41(p9,p10); //name the serial port rn41

BusOut led(LED4,LED3,LED2,LED1);

int main() {

rn41.baud(115200); // setup baud rate

rn41.putc('$'); // Enter command mode

rn41.putc('$');

rn41.putc('$');

wait(1.2);

rn41.putc('S'); //set Authentication to 0

rn41.putc('A');

rn41.putc(',');

rn41.putc('0');

rn41.putc(0x0D);

rn41.putc('-'); // Exit command mode

rn41.putc('-');

rn41.putc('-');

rn41.putc(0x0D);

wait(0.5);

while (1) {

if (rn41.readable()) { // if data available

char x=rn41.getc(); // get data

led=x; // output LSByte to LEDs

}

}

}

Program Example 11.4: Bluetooth Slave program

/\* Program Example 11.5: Zigbee serial test data program

Data is transferred from mbed to PC via Zigbee.

Requires a set of "paired" XBee modules. \*/

#include "mbed.h"

Serial xbee(p9,p10); //name the serial port xbee

BusOut led(LED4,LED3,LED2,LED1);

int main() {

xbee.baud(9600); // set baud rate for xbee

while (1) {

for (char x=0x30;x<=0x39;x++){ // ASCII numerical characters 0-9

xbee.putc(x); // send test char data on serial to XBee

led = x & 0x0F; // set LEDs to count in binary

wait(0.5);

}

}

}

Program Example 11.5: Zigbee serial test, mbed to PC

/\* Program Example 11.6: Zigbee

Data is transferred bidirectionally between mbed and PC via Zigbee \*/

#include "mbed.h"

Serial xbee(p9,p10); //set up the serial port and name xbee

BusOut led(LED4,LED3,LED2,LED1);

int main() {

xbee.baud(9600); // set up baud rate

xbee.printf("Serial data test: outputs received data to LEDs\n\r");

while (1) {

if (xbee.readable()) { // if data available

char x=xbee.getc(); // get data

led=x; // output LSByte to LEDs

}

}

}

Program Example 11.6: Zigbee bidirectional data test

/\* Program Example 11.7

Paired Zigbees - coordinator program

Requires a paired set of XBees, configured in XCTU

Hardware: XBee and mbed located in app board \*/

#include "mbed.h"

#include "C12832.h"

C12832 lcd(p5, p7, p6, p8, p11);

Serial xbee(p9,p10); //name the serial port xbee

BusOut led(LED4,LED3,LED2,LED1);

char x,j;

int main(){

lcd.cls(); //clear lcd screen

lcd.locate(0,3); //locate the cursor

lcd.printf("Zigbee Test Program");

wait (1);

while(1) {

if (xbee.readable()){ // if data available

j=0;

x=xbee.getc(); // get data

led=x; // output LSByte to LEDs

lcd.locate(0,15);

lcd.printf("Remote data = %d",x);

}

else { //count no of times there is no data

j++;

wait (0.01);

}

if (j>250){

lcd.locate(0,15);

lcd.printf("Wireless link lost!");

j=0; //reset counter

}

}

}

Program Example 11.7: XBee to Xbee link - coordinator

/\* Program Example 11.8

Paired Zigbees - router program

The router generates data and sends to coordinator \*/

#include "mbed.h"

Serial xbee(p9,p10); //name the serial port xbee

BusOut led(LED4,LED3,LED2,LED1);

char x;

int main() {

xbee.baud(9600);

while (1) {

x++; // increment x

if (x>0x0F) // limit to 4 bits

x=0;

xbee.putc(x); // send char data on serial

led = x; // set LEDs to count in binary

wait(0.5);

}

}

Program Example 11.8: XBee to Xbee link - router

/\* Simple XBee API application

Requires XBee and mbed in app board, plus remote XBee

The coordinator receives data from router, and displays on lcd \*/

#include "mbed.h"

#include "C12832.h"

C12832 lcd(p5, p7, p6, p8, p11);

Serial xbee(p9,p10); //name the serial port xbee

DigitalOut led (LED4);

char x,xhi,xlo,j,len,ftype; //some useful internal variables

int result;

int main(){

lcd.cls(); //clear lcd screen

lcd.locate(0,3); //locate the cursor

lcd.printf("Zigbee API Test");

wait (1);

while(1) {

if (xbee.readable()) // if data is available

x=xbee.getc(); // get data

if (x==0x7E){ //test for start of frame

led=1; //set diagnostic LED

while (xbee.readable()==0); //wait for next byte

x=xbee.getc(); //discard length msb, assume zero

while (xbee.readable()==0);

len=xbee.getc(); //save length lsb

while (xbee.readable()==0);

ftype=xbee.getc(); //save frame type

j=1;

//now discard 15 bytes: i.e. 64 bit address, 16-bit address, et al

while(j<16){

while (xbee.readable()==0); //wait

x=xbee.getc(); //discard

j++;

}

while (xbee.readable()==0);

xhi=xbee.getc(); //get ms ADC byte

while (xbee.readable()==0);

xlo=xbee.getc(); //get ls ADC byte

result = xhi\*256 + xlo; //convert to 16 bit number

lcd.locate(0,15);

lcd.printf("length = %d",len); //include values as desired

lcd.printf(" data = %d",result); //display result

led = 0;

} //end of if

} //end of while(1)

}

Program Example 11.9: Applying the XBee API – coordinator program