Bluetooth

Podstawy

- Technologia do komunikacji bezprzewodowej bliskiego zasięgu
- Komunikacja typu master - slave
- Standard IEEE 802.15.1
- Pierwsza wersja specyfikacji (v1.0) - 1999

Rozwój interfejsu

2007

- V2.1 + EDR - Secure Simple Pairing allows secure device pairing with a button press, numeric entry, numeric compare, and Out of Band

2009

- V3.0 + HS - High Speed Enables applications to use 802.11 MAC/PHY through addition of Generic Alternate MAC/PHY

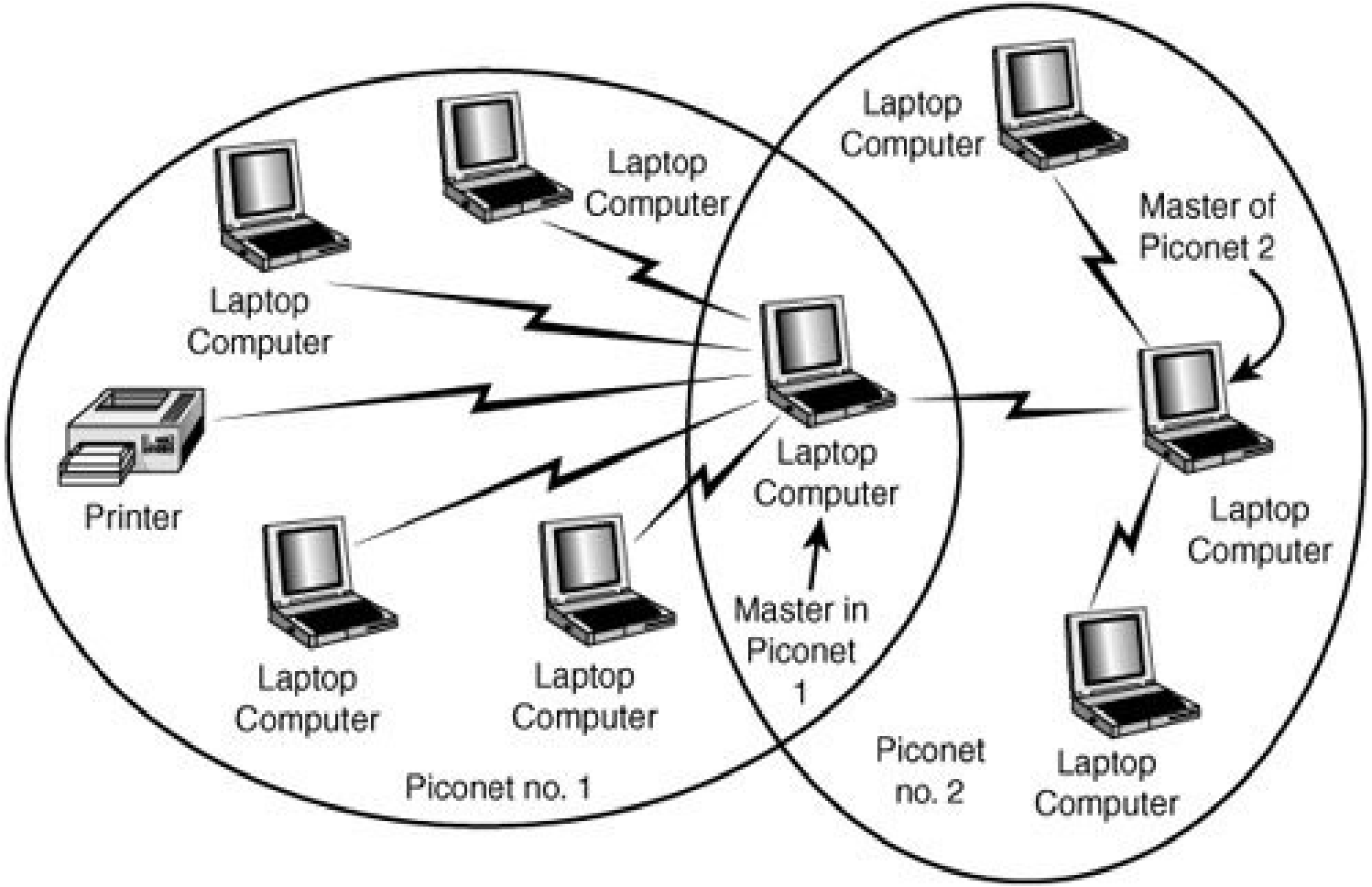
2010

- V4.0 - Low Energy Enables new applications in different markets including healthcare, sports/fitness, security, home entertainment

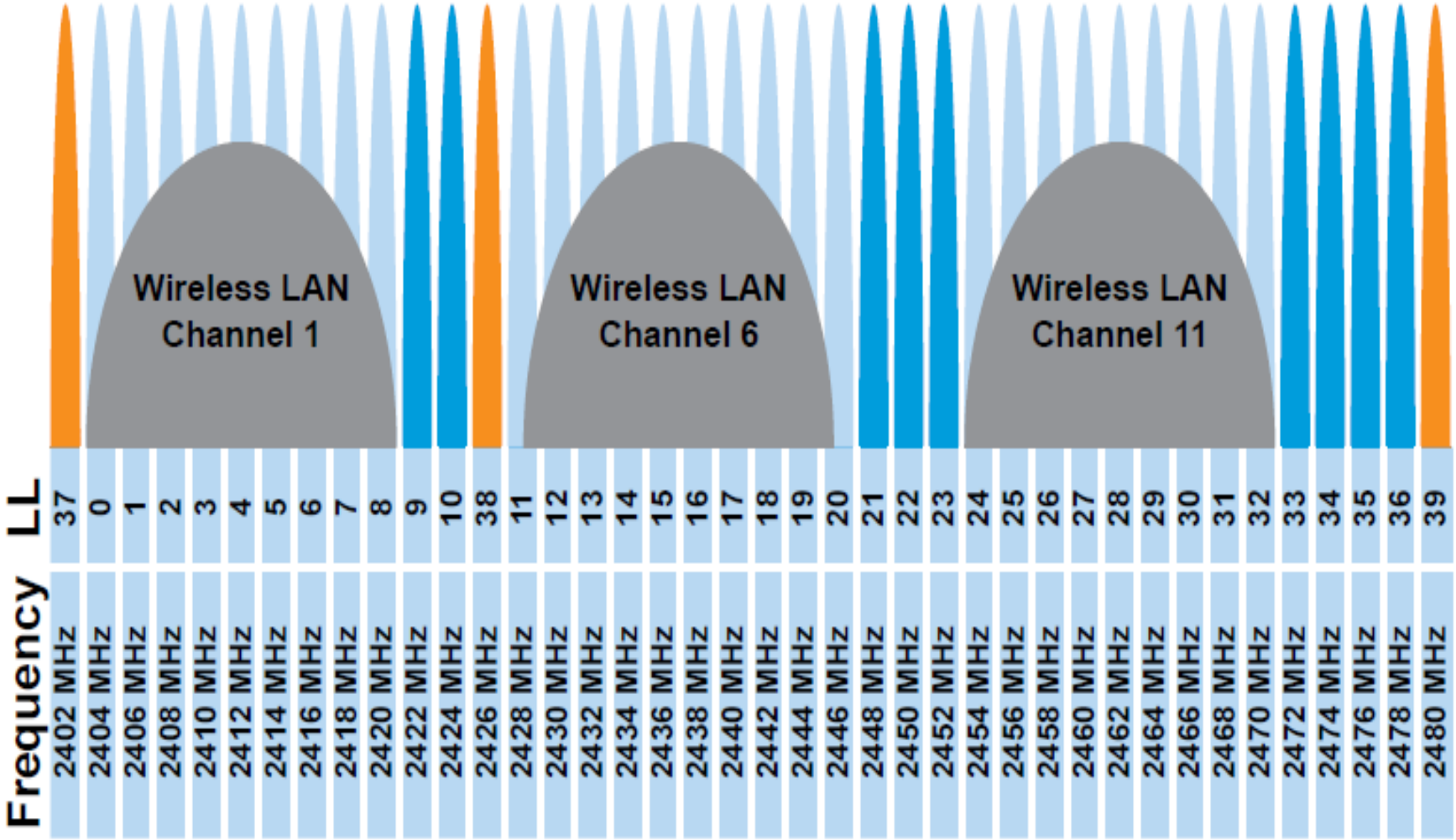
Parametry

Technology	*Bluetooth* BR/EDR/HS Technology	*Bluetooth* Low Energy Technology
Radio Frequency	2.4 GHz ISM	2.4 GHz ISM
Range	10 to 100 meters	10 to 100+ meters
Data Rate	1-3 Mbps (Classic) >400 Mbps (AMP, 802.11n)	1 Mbps
Nodes/Active Slaves	7 / 16777184	Unlimited
Security	56b E0 (classic)/128b AES (AMP) and applications layer user defined	128b AES and application layer user defined
Robustness	Adaptive frequency hopping, FEC	Adaptive frequency hopping
Latency (from non connected state)	100ms	<3ms
Regulatory Acceptance	Worldwide	Worldwide
Voice Capable	Yes	No
Network Topology	Scatternet	Star-bus
Power Consumption	1 as the reference, x10 for AMP	0.01 to 0.5 (use case dependent)
Service Discovery	Yes	Yes

Architektura sieci



Współdzielenie pasma transmisyjnego



Uprawnienia

```
<uses-permission android:name =
                    "android.permission.BLUETOOTH" />

<uses-permission android:name =
                    "android.permission.BLUETOOTH_ADMIN"/>
```

Włączenie Bluetooth

```
BluetoothAdapter btAdapter = BluetoothAdapter.getDefaultAdapter();

if(btAdapter != null) {
  if (!btAdapter.isEnabled()) {
    Intent enableBtIntent = new
        Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
}

if(btAdapter != null) {
    if (!btAdapter.isEnabled()) {
        btAdapter.enable();
}
```

Czas na wykrycie

```
Intent discoverableIntent = new
    Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
discoverableIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_
    DURATION, 0);
startActivity(discoverableIntent);
```

Unlimited discovery time

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zestawienie połączenia

```
final String nxtMACadress = "00:16:53:0E:5E:14";

BluetoothDevice nxtDevice = btAdapter.getRemoteDevice(nxtMACadress);
try {
    BluetoothSocket nxtSocket = null;
    nxtSocket = nxtDevice.createRfcommSocketToServiceRecord(
    UUID.fromString("00001101-0000-1000-8000-00805F9B34FB"));
    try {
        nxtSocket.connect();
    } catch (IOException e) {
    e.printStackTrace();
}
```

Sterowanie NXT

```
InputStream nxtInputStream = nxtSocket.getInputStream();
OutputStream nxtOutStream = nxtSocket.getOutputStream();

byte message[] = new byte[12];
int motor = MOTOR_A;
int speed = 50;
int messageLength = message.length;

byte DIRECT_COMMAND_NOREPLY = 0x80;
byte SET_OUTPUT_STATE = 0x04;
int MOTOR_A = 0;
int MOTOR_B = 1;
int MOTOR_C = 2;
```

Sterowanie serwonapędem

```
message[0] = DIRECT_COMMAND_NOREPLY;
message[1] = SET_OUTPUT_STATE;
message[2] = (byte)motor;
message[3] = (byte)speed; // Range: -100 to 100
message[4] = 0x03; //MOTORON + BREAK
message[5] = 0x01; // Regulation mode
message[6] = 0x00;  // Turn Ratio
message[7] = 0x20; // RunState
message[8] = 0; // TachoLimit = run forever
message[9] = 0;
message[10] = 0;
message[11] = 0;
```

Wysłanie rozkazu

```
    nxtOutStream.write(messageLength);
    nxtOutStream.write(messageLength>>8);
    nxtOutStream.write(message,0,messageLength);
} catch (IOException e) {
e.printStackTrace();
}
```