Laborator 8 – Using the Bluetooth communication

1. Bluetooth basics

Bluetooth is a standardized communication protocol used for data communication based on the radio frequency band of 2.4 GHz. This frequency does not require an operating license and it is used by other wireless communication protocols like ZigBee or WiFi. Bluetooth technology is present in devices like smartphones, laptops, headphones, video game controllers, or portable spekers.

Fig. 1. Bluetooth: ideal for blasting your favorite music, for controlling robots with your tablet, or for cheating at the exams[1].

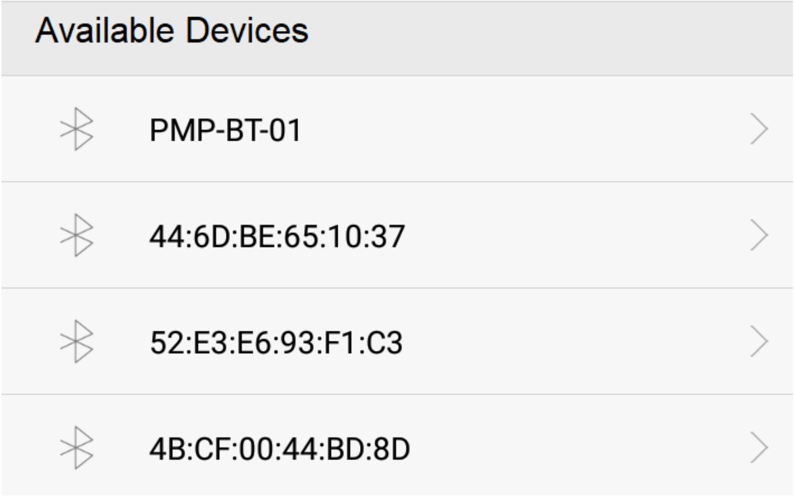
Bluetooth networks (also called piconets) use a master/slave arrangement to control when and where the devices can exchange information. A master device can be connected to up to 7 slave devices, but one slave device can only have connection to one master. The master device coordinates the communication in the network, being able to send data to any slave or to request information from any slave. Slave devices can only communicate with the master device, not with other slave devices from the network.

1.1 Names and addresses

Every Bluetooth device has a unique address of 48 bits, named BD_ADDR. This address is usually visible in the form of a 12 digits number, in hexadecimal format. The first 24 bits identify the device producer and the last 24 represent a unique address. Sometimes we can find this address named as “MAC Number” or “MAC address”.

Bluetooth devices can also have a name which is easier to remember. If this name exists, it is usually displayed on the master device’s screen, when searching for new devices to form its network. The names can have up to 248 of characters in length, and multiple devices can share the same name. This situation can cause problems in a lab environment, where multiple Bluetooth devices can be active in the same time. For this reason we have made sure that every device has a unique name, of the pattern **PMP-BT-XX,** where XX is a 2 digit number ranging from 01 to 15.

[1] Please don’t cheat at the exams, because you will get caught and punished according to the ECTS rules.



Available Devices

PMP-BT-01

44:6D:BE:65:10:37

52:E3:E6:93:F1:C3

4B:CF:00:44:BD:8D

Fig. 2. Bluetooth devices found by accessing the Bluetooth settings of a mobile phone. The first identified device is found based on its name, and the rest are found based on their 12 hexadecimal digits address. It is possible for a device to be first identified based on the address, and the name to appear later.

1.2 Energy classes

The power of energy transmission of a Bluetooth device is defined by its energy class. There are 3 energy classes. In table I we see the different classes of energy.

Table I. Classes of energy and their maximum power

Class Number	**Maximum Power (mW)**	**Maximum Distance**
Class 1	100 mW	100 m
Class 2	2.5 mW	10 m
Class 3	1 mW	10 cm

Some modules can operate in a single energetic class, and others can vary their transmition power.

1.3 The connection process

Connecting two Bluetooth devices is a process which requires several steps:

1. Interrogation (discovery): If two devices do not know anything about the other, one will try to find the existence of the other. One of the devices (usually a laptop, phone or tablet) will send a discovery request and any device listening and it is in the active range will respond with an address, name and other information. The addresses and names from figure 2 are displayed after the discovery process.

2. Pairing (Binding): The pairing process is realized only once for a pair of devices, and it implies authentication, the user being requested to validate the connection between the two devices. Sometimes the validation only requires the press of a button, and other times it requires the input of a 6 or 4 digit code. For the devices used in the lab the PIN code is 1234.

When two devices are paired, they store the address of other device as well as its name and profiles. Also, they will share a secret key which will permit them to connect automatically in the future and will ensure the security of the data transmission.

3. Connection: after pairing, the actual connection can take place. During the connection phase data can be sent and received among the devices (Active mode), or the devices can enter a waiting phase, to lower the energy consumption.

2. The HC-05 Bluetooth module

HC-05 is a module designed to be easy to use in Arduino projects. Its implicit configuration is in Slave mode, having the name HC-05 and serial interface settings compatible with the default settings of the Arduino serial interface settings (Baud Rate 9600, No parity, 8 data bits, 1 stop bit). The module can be purchased in the 4 pin version (VCC, GND, RX și TX), and in the 6 pin version, which allows device configuration and program-based state checking.

Technical specifications:

- Supply voltage: 3.6V - 6V
- Energy Consumption: 30mA
- Range: max. 100m
- It uses the IEEE 802.15.1 standardized protocol
- It can be operated as master or slave.
- Baud Rate: 9600, 19200, 38400, 57600, 115200, 230400, 460800

The available pins for the devices in the lab are:

1. EN (sometimes with the name Key): If it is connected to logic level 1 before powering the module, the device enters the configuration mode (the LED blinks slowly, once every 2 seconds). In this mode the module waits for AT type commands from the serial interface, at a baud rate of 38400 bits per second. **For normal operation this pin should not be connected**!
2. VCC: +5v – supply voltage pin, should be connected to a 5V Arduino pin.
3. GND: Ground pin, should be connected to a GND Arduino pin.
4. TXD: Transmission pin of the serial interface, should be connected to an RXn pin of the Arduino board (RX1 for this lab work).
5. RXD: Reception pin of the serial interface, should be connected to a TXn pin of the Arduino board (TX1 for this lab work).
6. STATE: it tells us if the device is connected or not. This pin is connected to the LED from the module. Based on the operating mode the led blinks faster or slower:
   - Blinks once each 2 seconds: the module is in configuration (command) mode.
   - Fast blink: the module is waiting for connections- **this mode should be active at system start**.
   - Blink twice each second. The connection has been established and we can send and receive data.

In figure 3 we can see an image of the module and its pins.

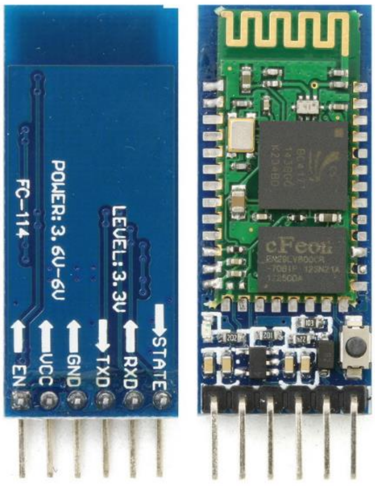


Fig. 3. The HC-05 module and its pins.

3. Connecting the Bluetooth module to Arduino

For the correct operation of the module in slave mode, as interface for data transmission, the module has to be connected as shown in figure 4. Connect the supply pins to 5V and GND, and the serial pins to the RX1 and TX1 pins corresponding to the hardware serial interface **Serial1**. For a more robust setup, use the BreadBoard: insert the pins of the module in it, and then make the connections with wires. **Pay close attention to the supply pins, the VCC pin is near the GND pin!**

After connecting the module to the Arduino Mega, after the module is powered its LED should start to blink fast, meaning that the module is waiting for a connection. **If the LED is not turning on, or it displays another working mode, turn off the system and inform the laboratory professor!**

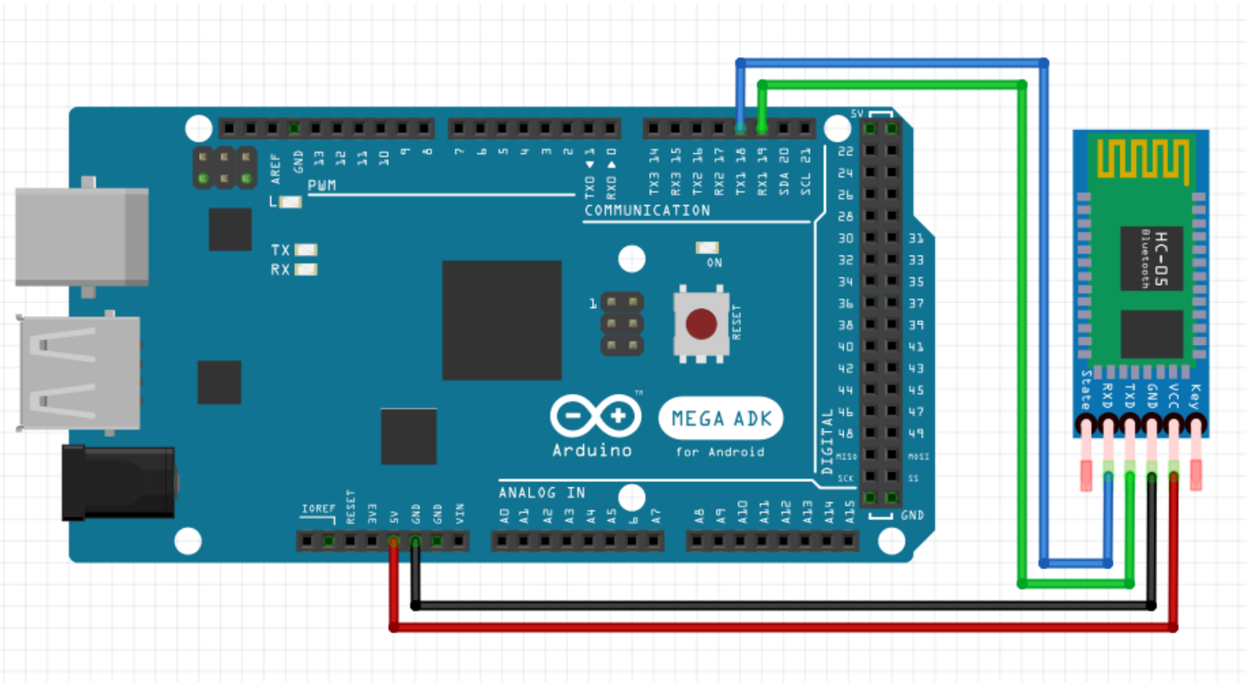


Fig. 4. Connection scheme of the HC-05 module to Arduino Mega.

3.1 The Arduino program for sending and receiving data

The following program will illustrate the bidirectional communication between Arduino and Bluetooth and Arduino and PC. Compile and upload the program bellow.

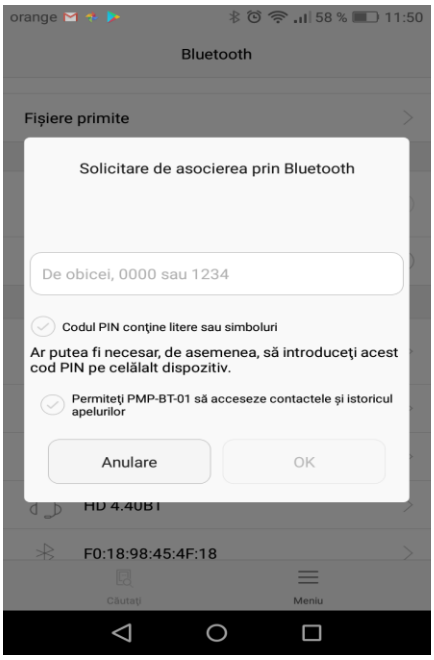
```
void setup()

{

  Serial.begin(9600);        // Serial 0 interface for PC

  Serial1.begin(9600);       // Serial 1 interface for Bluetooth module

}


void loop()

{

  if (Serial1.available()) // Read from Bluetooth and send to PC

    Serial.write(Serial1.read());

  if (Serial.available())  // Read from PC and send to Bluetooth

    Serial1.write(Serial.read());

}
```

Now Arduino is ready to communicate in a wireless mode with another device. For illustrating such a communication, we will need a mobile phone and a Bluetooth terminal program. Such a program is Serial Bluetooth terminal, which can be found on Google Play (reference 4).

3.2 Running the example program

Follow the steps bellow to establish a connection between the Arduino and the mobile phone:

1. Download the Serial Bluetooth Terminal, or an equivalent application, on your phone.
2. Connect the Bluetooth module to the Arduino board, based on the instructions given above, and program the Arduino board with the given program.
3. Activate the Bluetooth function on your device and enter in the settings menu in the Bluetooth section, in order to discover the available devices.
4. Identify the name of your device. The name should follow the pattern PMP-BT-XX, the last two digits corresponding to the number written with the marker on your module.
5. Start the pairing operation with the device. If you are required a PIN, enter 1234. In figure 5 we can see the pairing operation.



Solicitare de asocierea prin Bluetooth

De obicei, 0000 sau 1234

○ Codul PIN conţine litere sau simboluri

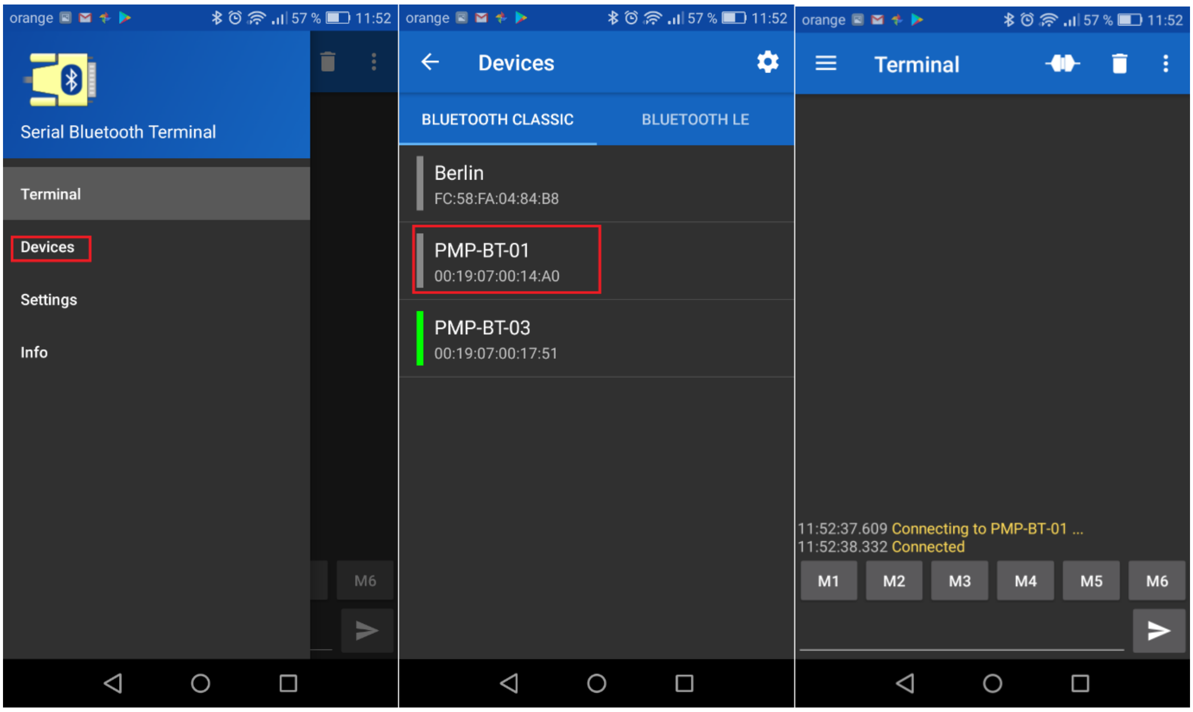
Ar putea fi necesar, de asemenea, să introduceţi acest cod PIN pe celălalt dispozitiv.

○ Permiteţi PMP-BT-01 să acceseze contactele şi istoricul apelurilor

Anulare OK

Fig. 5. Pairing.

6. Start the mobile phone app. From the application menu select „Devices" and then select your device. In this moment the connection should establish. This process is depicted in figure 6.



Berlin
FC:58:FA:04:84:B8

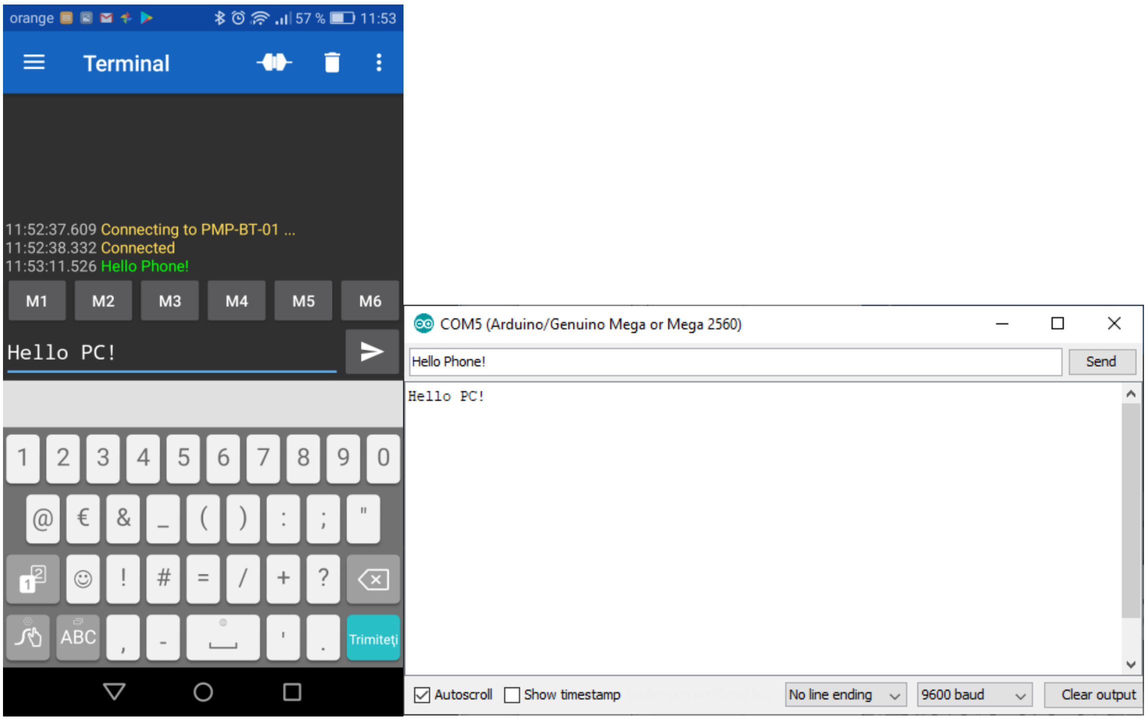
PMP-BT-01
00:19:07:00:14:A0

PMP-BT-03
00:19:07:00:17:51

```
11:52:37.609 Connecting to PMP-BT-01 ...
11:52:38.332 Connected
```

Fig. 6. Establishing the connection

7. On the PC, open the Serial Monitor. Write characters in the program and press Send. The characters should appear on the mobile phone. You can also type characters on your mobile phone and the characters will appear on the serial monitor.



```
11:52:37.609 Connecting to PMP-BT-01 ...
11:52:38.332 Connected
11:53:11.526 Hello Phone!
```

Hello PC!

COM5 (Arduino/Genuino Mega or Mega 2560)

Hello Phone!

```
Hello PC!
```

☑ Autoscroll ☐ Show timestamp No line ending 9600 baud

Fig. 7. An example of message exchange between the PC and the mobile phone, via Arduino

4. Independent work

1. Run the example, carefully following the steps described in the laboratory work.
2. Write a program that will transmit through Bluetooth, to the mobile phone, the value of the temperature read from a sensor (once every second). The program will receive from the mobile phone commands that will be interpreted such that the temperature transmission rate will be increased or decreased (for example, the command „r+" will increase the transmission rate and the command „r-" will decrease the transmission rate).
3. Measure the blink rate of the module's LED, by reading the sixth pin of the Bluetooth module. Based on this rate, detect the status of the Bluetooth module (waiting for connections, connection established, etc), and display this state in the Serial Monitor.
4. Connect two Arduino boards through I2C. Each board should also be equipped with a Bluetooth module. Make a bidirectional communication between two phones using Arduino as a proxy.

References

1. Bluetooth Basics: https://www.ele.uri.edu/courses/bme362/handouts/Bluetooth.pdf

2. Modul Bluetooth HC-05 (160) compatibil Arduino http://roboromania.ro/produs/modul-bluetooth-hc-05-compatibil-arduino/

3. HC-05 Datasheet http://roboromania.ro/datasheet/Bluetooth-HC-roboromania.pdf

4. Serial Bluetooth Terminal,

https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal&hl=en