Lab 2: Communication between IoT devices

Objective: the main objective of this exercise is to understand the communication between IoT devices (in this case, Arduino), including encoding the data, using a checksum and input/output devices.
The report must include **ALL** points mentioned in the exercises' descriptions.
Format for the report name: "Lab_2_*surname1*_*surname2*.pdf"

1 Description

The task is to connect two Arduinos (Figure 1) using one of the interfaces (SPI, I2C or Serial) for transceiving control commands, which control an output device.
The first Arduino should receive symbol commands from a Virtual Terminal (Figure 2), encode them to byte packages, calculate checksums of each package, and send the packages and checksums to the second Arduino.

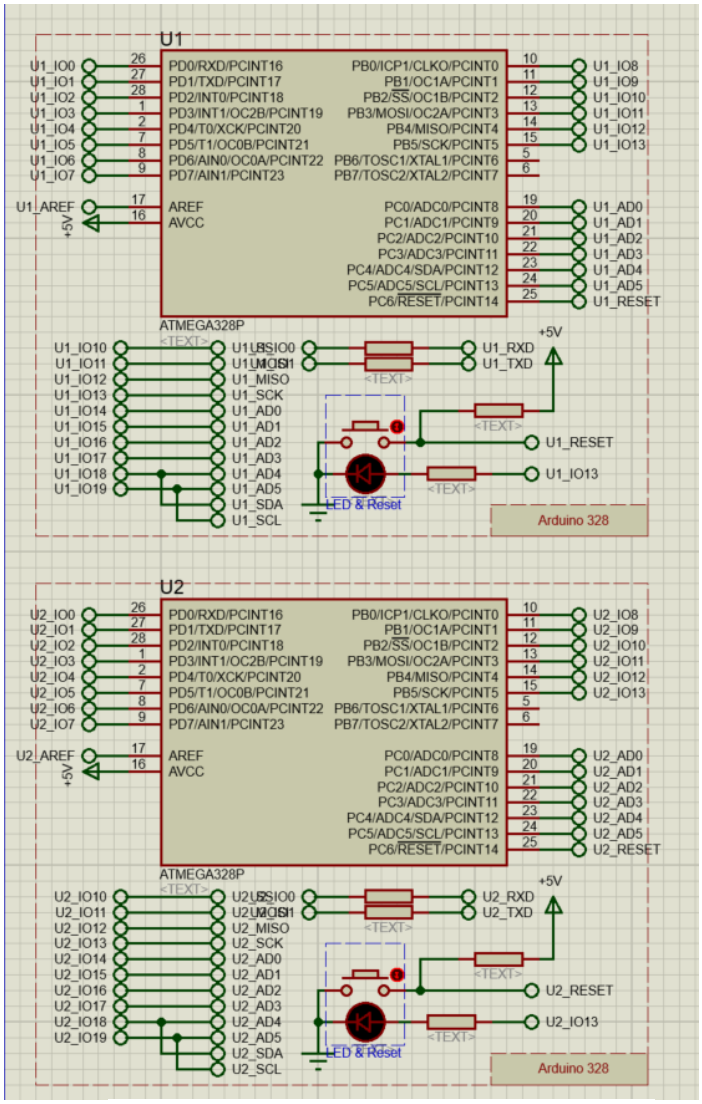

Figure 1. Arduino simulation model

Checksum

A checksum is a sequence of numbers and letters used to check data for errors. You can use a checksum utility to confirm your copy is identical if you know the checksum of an original file. You can use checksums to check files and other data for errors that occur during transmission or storage. More about checksum and cyclic redundancy check you can find the following links:

- https://en.wikipedia.org/wiki/Cyclic_redundancy_check,
- https://users.ece.cmu.edu/~koopman/pubs/KoopmanCRCWebinar9May2012.pdf
- https://barrgroup.com/tech-talks/checksums-and-crcs

Virtual terminal

Virtual Terminal is a tool in Proteus, which is used to view data coming from Serial Port and send it to Serial Port. It is a very easy-to-use tool that requires only two wires to communicate: RX and TX. In order to see what you type on Virtual Terminal, you have to enable the **echo typed characters** option, which can be found in the right-click menu of the terminal window. There is also a paste option on that menu. How to use the Virtual Terminal and more information about it you can find at these links on the Proteus help window:

- https://pic-microcontroller.com/how-to-use-virtual-terminal-in-proteus-isis/,
- https://www.electronicslovers.com/2015/04/how-to-use-virtual-terminal-in-proteus.html) or

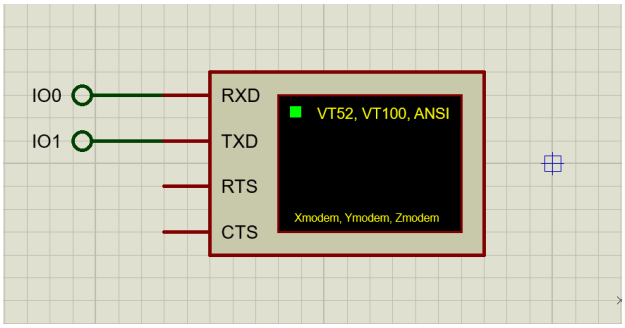

Figure 2. Virtual Terminal

Encoding

How to encode symbol commands to byte packages you should decide by yourself. You can create your own encoding-decoding rules, where each symbol command will correspond to a specific set of bytes. For example: the command "LED_ON" corresponds the bit packet "0000 0001 + checksum", the command "LED_OFF" corresponds the bit packet "0000 0011 + checksum, etc. The length of each packet must be constant and at least 8 bits. Each student group will have different commands according to the different output devices.

The second Arduino should receive data from the first and check the received data for lost or manipulated contents (using checksums). If part of the package is lost, the package should be skipped. If the packages are correct, they should be decoded into corresponds output states of the output device connected to the Arduino. The block diagram is shown in Figure 3.

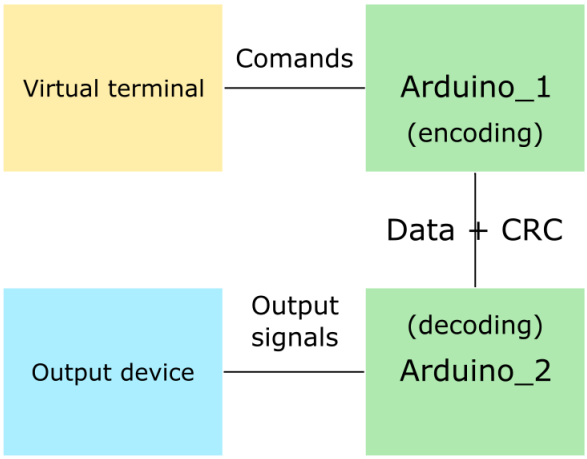

Figure 3. System structure

General task

In Moodle you can find the template-project with 2 Arduinos and Virtual Terminal. Start this project, check input/output ports of Arduino devices, and work with the Virtual Terminal. Think about the possibilities of connecting I/O devices and try one of the following tasks:

- connect and change the colour of RGB Led to blue,
- connect and rotate the servo motor 360 degrees in one direction,
- connect and display letter “U” on the Led Matrix.

Important: At this point it is not necessary to connect Arduinos to each other. The exact names of output devices in Proteus can be found below in the description of individual tasks.

You should demonstrate to one of the tutors the work of the implemented function.

2 Individual tasks

Each student group has a different combination of communication interface, checksum type, and output device, according to Table 1.

Table 1. Variants of tasks

Group number	Communication interface	Checksum type	Output device
1	SPI	XOR	LED Matrix
2	I2C	CRC-8	RGB Led
3	Serial	CRC-16	Servo motor
4	SPI	CRC-8	Servo motor
5	I2C	CRC-16	LED Matrix
6	Serial	XOR	RGB Led
7	SPI	CRC-16	RGB Led
8	I2C	XOR	Servo motor
9	Serial	CRC-8	LED Matrix

There are three variants of output devices available:

- **For RGB Led** (Figure 4) at least five commands should be implemented:
  1. “set yellow color”;
  2. “set violet color”;
  3. “set white color”;
  4. "blink alternately" - blink alternately 7 colors of a rainbow;
  5. “off”.

  The color shades do not matter.

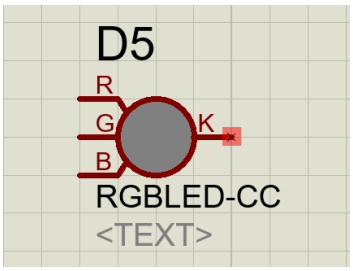

Figure 4. RGB Led simulation model

- **For LED Matrix** (Figure 5) at least two commands should be implemented:
  1. “set number <0 – 9>” - displays numbers from 0 to 9 on the matrix, respectively.
  2. “off”

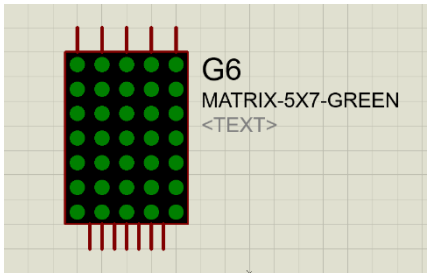

Figure 5. LED Matrix simulation model

- **For Servo motor** (Figure 6) at least three commands should be implemented:
  1. “rotate <x> degrees right”
  2. “rotate <x> degrees left”
  3. “wave” - Periodic turns of 180 degrees to the right and back.

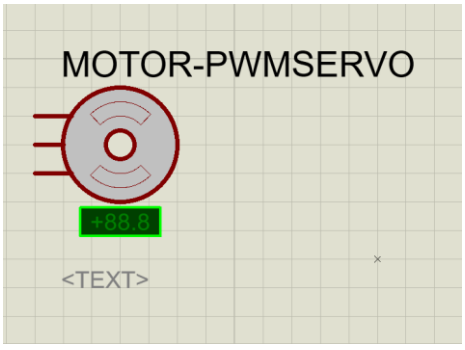

Figure 6. Servo motor simulation model

Important notice:
The schematic is not complete, and one task is that you should complete it by adding additional components according to your personal task. You can find how to connect different input/output devices to Arduino on the Internet. You may not use any libraries for calculating the checksum, encoding, or decoding data.
You can use additional libraries for communication interfaces and a servo motor. For student groups that are planning to use the Serial interface, it is recommended to use a software serial port (https://www.arduino.cc/en/Reference/softwareSerial) for a stable simulation process. You also can use an additional Virtual Terminal connected to the second Arduino for debugging.

3 Report requirements

In your report, you must:

1. Describe in detail each step of your work. Attach screenshots of the circuit you have assembled. Shortly describe the connections that you made and possibly additional components and their purpose.
2. Describe your encoding-decoding algorithm, attach the corresponding code.
3. Describe the algorithm for calculating the control sum, attach the corresponding code.
4. Describe the control commands that you created.
5. Shortly describe your output device, its properties, and control principles.
6. Attach some screenshots of a running system. We should see the Virtual Terminal commands that you enter and the output states of the output device in accordance with these commands.
7. Upload your report in Moodle (format for the report name: “Lab_2_surname1_surname2.pdf”, it is enough, if only one person from the group will upload the report) and present the system work to the tutors. All members of working group should be able to explain the work and answer the questions!