Ministry of Education, Culture and Research of the Republic of Moldova

Technical University of Moldova

Department of Software and Automatic Engineering

**REPORT**

Laboratory Project nr.6

*at Embeded Systems*

Done by:

Gr.172-FAF Ion Dodon

Checked by: Andrei Bragarenco

Chișinău 2020

Laboratory Project Nr.5

**Topic:** Comunicare

**Objective:** The objective of this laboratory work is to understand how transmit data between devices as kind of implemented protocol.

**Domain: T**he [Internet of Things (IoT)](https://www.getkisi.com/technologies/internet-of-things-iot), is based on the networking of things. In a nutshell, Internet of Things is defined as a [“proposed development of the Internet in which everyday objects have network connectivity, allowing them to send and receive data.”](http://www.gsma.com/connectedliving/gsma-vision-of-smart-home-report/)

**The most important thing here is connectivity among objects.**

We can boil down the wireless communication protocols into the following 6 standards:

* Satellite
* [WiFi](https://www.getkisi.com/technologies/wireless-access-control)
* Radio Frequency (RF)
* [RFID](https://www.getkisi.com/technologies/rfid-access-control)
* [Bluetooth](https://www.getkisi.com/guides/bluetooth-access-control)
* [NFC](https://www.getkisi.com/technologies/nfc-access-control)

**Component description**:

**Electric lamp** - An **electric lamp** is a conventional light emitting component used in different circuits, mainly for lighting and indicating purposes. The construction of lamp is quite simple, it has one filament surrounding which, a transparent glass made spherical cover is provided. The filament of the lamp is mainly made of tungsten as it has high melting point temperature. A lamp emits light energy as the thin small tungsten filament of lamp glows without being melted, while [current](https://www.electrical4u.com/electric-current-and-theory-of-electricity/) flows through it.

**Arduino Uno -** The **Arduino Uno** is an [open-source](https://en.wikipedia.org/wiki/Open-source) [microcontroller board](https://en.wikipedia.org/wiki/Microcontroller_board) based on the [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) microcontroller and developed by [Arduino.cc](https://en.wikipedia.org/wiki/Arduino).[[2]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-2)[[3]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-What_is_Arduino?-3) The board is equipped with sets of digital and analog [input/output](https://en.wikipedia.org/wiki/Input/output) (I/O) pins that may be interfaced to various [expansion boards](https://en.wikipedia.org/wiki/Expansion_board) (shields) and other circuits.[[1]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-Makerspace-1) The board has 14 digital I/O pins (six capable of [PWM](https://en.wikipedia.org/wiki/Pulse-width_modulation) output), 6 analog I/O pins, and is programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment), via a type B [USB cable](https://en.wikipedia.org/wiki/USB_cable).[[4]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-priceton-4) It can be powered by the USB cable or by an external [9-volt battery](https://en.wikipedia.org/wiki/9-volt_battery), though it accepts voltages between 7 and 20 volts. It is also similar to the [Arduino Nano](https://en.wikipedia.org/wiki/Arduino_Nano) and Leonardo.[[5]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-5)[[6]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-6) The hardware reference design is distributed under a [Creative Commons](https://en.wikipedia.org/wiki/Creative_Commons) Attribution Share-Alike 2.5 license and is available on the Arduino website. Layout and production files for some versions of the hardware are also available.

**Liquid Crystal Display**: A **liquid-crystal display** (**LCD**) is a [flat-panel display](https://en.wikipedia.org/wiki/Flat_panel_display) or other [electronically modulated optical device](https://en.wikipedia.org/wiki/Electro-optic_modulator) that uses the light-modulating properties of [liquid crystals](https://en.wikipedia.org/wiki/Liquid_crystal) combined with [polarizers](https://en.wikipedia.org/wiki/Polarizer). Liquid crystals do not emit light directly,[[1]](https://en.wikipedia.org/wiki/Liquid-crystal_display#cite_note-1) instead using a [backlight](https://en.wikipedia.org/wiki/Backlight) or [reflector](https://en.wikipedia.org/wiki/Reflector_(photography)) to produce images in color or [monochrome](https://en.wikipedia.org/wiki/Monochrome).[[2]](https://en.wikipedia.org/wiki/Liquid-crystal_display#cite_note-2) LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and [seven-segment displays](https://en.wikipedia.org/wiki/Seven-segment_display), as in a [digital clock](https://en.wikipedia.org/wiki/Digital_clock). They use the same basic technology, except that arbitrary images are made from a matrix of small [pixels](https://en.wikipedia.org/wiki/Pixel), while other displays have larger elements. LCDs can either be normally on (positive) or off (negative), depending on the polarizer arrangement. For example, a character positive LCD with a backlight will have black lettering on a background that is the color of the backlight, and a character negative LCD will have a black background with the letters being of the same color as the backlight. Optical filters are added to white on blue LCDs to give them their characteristic appearance.

**Implementation:**

[**https://drive.google.com/open?id=1-6iEo9zzrN4-zUwgVG2uP-Ac914HL\_ZN**](https://drive.google.com/open?id=1-6iEo9zzrN4-zUwgVG2uP-Ac914HL_ZN)

**Master:**

#include "../include/Lab1/libs.h"

#include "../include/Lab1/conts.h"

Lcd \*lcd;

MySerial \*serial;

StaticJsonDocument<200> packet;

int distance;

char receive[5];

int nr = 1;

int receiverId = 4;

char command[10];

char request[100];

int x = 0;

void receiveEvent(int);

void setup() {

  lcd = new Lcd(lcdPinOne, lcdPinTwo, lcdPinThree, lcdPinFour, lcdPinFive, lcdPinSix);

  serial = new MySerial();

  Wire.begin(address);

  Wire.onReceive(receiveEvent);

  lcd->openStream();

}

void loop() {

  delay(500);

  if(x % 3 == 0){

    strcpy(command, "blink\_led");

  } else {

    strcpy(command, "get\_data");

  }

  int size = sizeof(command);

  int checksum = size \* 2;

  packet["start"] = "STX";

  packet["pkID"] = nr++;

  packet["data"] = command;

  packet["cksum"] = checksum;

  serializeJson(packet, Serial);

  memset(command, 0, sizeof(command));

  delay(500);

  if(serial->hasMessage()){

    delay(100);

    int i = 0;

    while(serial->hasMessage()){

      char c = Serial.read();

      request[i] = c;

      i++;

    }

  }

  if(strncmp(request, "dist", 4) == 0){

  } else if (strcmp(request, "blink") == 0){

    lcd->setCursorLCD(0,0);

    printf("%s", request);

  }

  delay(200);

  memset(request, 0, sizeof(request));

  memset(receive, 0, sizeof(receive));

  x++;

  lcd->clearScreen();

}

void receiveEvent(int bytes){

  int i = 0;

  while(Wire.available()){

    receive[i] = Wire.read();

    i++;

  }

   lcd->setCursorLCD(0,0);

    printf("Result value:");

    lcd->setCursorLCD(0,1);

    printf("%s", receive);

}

**Slave:**

#include "../include/Lab1/libs.h"

#include "../include/Lab1/conts.h"

Lcd \*lcd;

MySerial \*serial;

Ultrasonic \*ultrasonic;

Led \*led;

DynamicJsonDocument doc(1024);

int distance;

char distArray[5];

const char\* command;

String request = "";

int checksum;

bool isValid = false;

char response[10];

int receiverId = 9;

int address = 4;

void checkCommand();

void validatePacket();

void setup() {

  lcd = new Lcd(lcdPinOne, lcdPinTwo, lcdPinThree, lcdPinFour, lcdPinFive, lcdPinSix);

  ultrasonic = new Ultrasonic(ultrasonicTrigger, ultrasonicEcho);

  serial = new MySerial();

  led = new Led(ledPin);

  Wire.begin();

  lcd->openStream();

}

void loop() {

  lcd->setCursorLCD(0,0);

  if(serial->hasMessage()){

    delay(500);

    while(serial->hasMessage()){

      char c = Serial.read();

      request += c;

    }

    validatePacket();

    delay(200);

      printf("Result");

      lcd->setCursorLCD(0,1);

      printf("%s", command);

      delay(400);

      lcd->setCursorLCD(0,1);

      printf("                ");

      checkCommand();

  }

  request = "";

  command = "";

  delay(100);

}

void validatePacket(){

  if(request.charAt(0) == '{' && request.charAt(request.length() - 1) == '}'){

      deserializeJson(doc, request );

      const char\* startPacket = doc["start"];

      const char\* endPacket = doc["end"];

      if(strcmp(startPacket, "STX") == 0 ){

        command = doc["data"];

        checksum = doc["cksum"];

        int sizeOfCommand = strlen(command) + 2;

        if(checksum / 2 == sizeOfCommand){

          isValid = true;

        }

      }

    }

}

void checkCommand(){

      if(strncmp(command, "get\_data", 9) == 0){

        Serial.write("value distance:");

        distance = ultrasonic->getDistance();

        sprintf(distArray, "%d", distance);

        Wire.beginTransmission(mcu2Address);

        Wire.write(distArray);

        Wire.endTransmission();

      } else if(strncmp(command, "blink\_led", 10) == 0){

        strcpy(response, "led on");

        Serial.write("led on");

        led->blink(500);

      }

}

**Conclusions:** Working on this laboratory work am was implemented a communication between two Microcontrollers through I2C interface.I have understood how to connect two Arduino Uno devices in order to communicate between them. Data was transmitted using an implement specific protocol.