ILLINOIS INSTITUTE OF TECHNOLOGY

Course: IT-T 492 – Embedded Systems and Reconfigurable Logic

Final Project: RFID Access and Attendance System

Bruno Pinto bpereira@hawk.iit.edu
Joao Calmon jfiguere@hawk.iit.edu
Vladimir Zivkovic vzivkovi@hawk.iit.edu

05/03/2016

Abstract

Nowadays, several institutions, including the Illinois Institute of Technology (IIT) uses paper to get attendance list in class. However, this way is becoming obsolete since the use of paper provides many expenses and waste. Our project goals to solve this situation using Arduino and the Radio Frequency Identification (RFID), which is already used in to others functions in IIT, to get the attendance in class.

The idea was to use the RFID reader to get the attendance list when the students uses their RFID cards when enter in class. The system keeps the list and send by email to the Professors when he/she uses his/her RFID card.

Keywords

RFID, Attendance, Access, Arduino, Embedded System

Contents

Abstract 2
Keywords 2
Introduction 4
About Embedded Systems 5
Steps needed to create Embedded System [8] 6
About the System 7
How it works? 8
Components 9
Arduino Uno 9
Arduino Mega 9
Arduino Ethernet Shield 10
RFID Reader 10
RFID Card 11
Real Time Clock 11
Battery 11
Costs 12
Advantages 13
Conclusion 14
Bibliography 15
A – Code for Attendance System 16
Appendix B – Server code (without SD card) 19
Appendix C – System implemented with Galileo (instead Arduino Mega) 25

Introduction

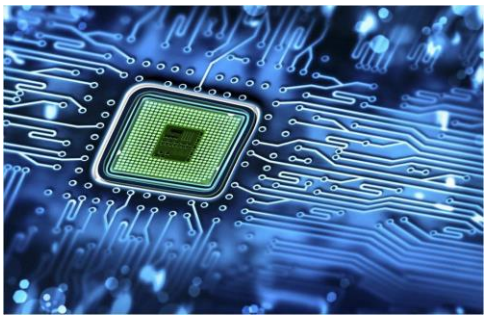
This paper goal to present our Final Project of the course **Embedded Systems and Reconfigurable Logic (IT-T 492)**, masterfully taught by professors **Jeremy Hajek** and **dr. Daniel R. Tomal**. The project summarize all the areas that we learned in class and in lab (Electricity, Data Collection, Data Transmission, and Data Presentation).

Our project is an attendance system that utilize the IIT Card of students to get their attendance in class replacing the attendance list paper. This system used the Arduino Uno as major component, since it is cheap, open source electronic and we learned how to use in several situations in class.

First, we will introduce you embedded systems and slowly move to the project. Second, we will present the components used to make this system possible. Third, we will explain how the components interact with each other in our system. Then, we will show the advantages of use our system instead use the actual system.

About Embedded Systems

Embedded Systems are present and future of our everyday life. Their possibilities are limitless. There are few reasons why we are extremely interested for them.



First reason what is fascinating about embedded systems for me is that they are applications which are **visible**. On the contrary of only software applications, embedded systems also have hardware part so you can see what you make! If you wrote code you can see physically how it works, what did you do! You can feel and touch embedded systems. Because your work is visible it can look like really high-tech, what can be impressive for you and anybody else!

Second reason why we are so impressed is that embedded systems can be **profitable**. If you have enough knowledge and enough luck, you can make money with them. There are **various applications** with embedded systems, and various possibilities. We can find them everywhere around. Every device with new features is possibility for embedded system. For example, we can make applications for smart homes, smart devices, etc. We can make application to turn off and on any device (for example, from our phone or PC). We can play with plenty sensors (distance sensors, light sensors, sensors for different kind of measurements, etc.). Our imagination can be limitless, and we can use it to make money.

One more reason, last but not least, is that interesting is also that for this you need to multi-skills in order to make this kind of systems. You need to be good in **hardware** as well as in **software**. From your basic idea, to design, program code, demo version and testing until final product there is a lot of work. But more interesting is that is not probably enough. You will need also to have skills to place it on the market! That is necessary to continue the process. You need to convince that somebody else find your product useful for self. And you need to be enough innovative and fast before anybody else offer same or similar product for less money. That is so complicated, but still very challenging and fun! And that is how you develop self and grow!

In the end, we can conclude that embedded systems give you possibility to make something new, some innovative thing! They give you possibility to make something visible, profitable and to develop your professional skills! They give you possibility to be creative. We tried to make one, which we will describe during this paper.

Steps needed to create Embedded System [8]

Because of certain application requirements, engineers have to approach embedded design in a different way than other types of designs.

The following offers a brief step-by-step approach to follow while designing an embedded system:

1. **Proposal:** - An innovative idea or system that makes life easier and/or reduces the amount of human effort required to complete a task.

2. **Definition**: - Next, the whole system needs to be designed, including what it will do under all possible sets of input conditions. This definition is perhaps the most critical part, as any error here will affect the working of whole system.

I. I/O Considerations: - Defines that for a particular input, what the output of the system will be considering the system as a black box.

II. Mathematical Modeling: - Design the algorithm for the system to work as desired.

III. Functional Modeling: - Design the functions of the system which will accept input and produce the desired output.

3**. Technology Selection**: - Based on the above points, designers then review available technology and select which devices will fulfill all the requirements while balancing efficiency, cost, and time-to-market.

4. **Integration & PCB design**: - List all the components, which you need to implement your functions and design their placement on the PCB. Traces and all other paths must have the least possible electromagnetic interference (EMI) and should be free from various errors. While designing the PCB, special attention must be given to the ground as well as all the components on the PCB that use ground.

5. **Firmware Development & Debugging**: - Since hardware needs instructions to execute the way we want, we need to write the code for every component used by the hardware. This is exactly what is done by the firmware i.e. the application code. Firmware should be of minimum complexity. Moreover, as we write the code, we face many errors or bugs and for this we need a proper debugging protocol.

6. **Testing**: Debugging tests the piece of code but in testing we test the whole system i.e. hardware as well as the software that drives that hardware.

7. **Documentation**: Anyone who accesses your complete application should never ask you “what does this mean?” or “How does this thing work?” and for this we need to document everything.

Now we will slowly move to the project we made. Ready?

About the System

Our product, RFID Access and Attendance system, have multi features such as: open/close the door - Access System, and sending real-time e-mail information for attendance (with students which attend the class/workers on the job/ researchers in the laboratory, and so on).

There are institutions which always need control of attendance their premises. For example, in IIT (although system can be used anywhere), there are premises such as Chemistry lab or Embedded System lab which would need this kind of system. As there are a lot of people which are using equipment, components and other resources, there is a need for this kind of system.

In the local level, customer can be IIT laboratories or IIT in general if the School decide to implement this system. In the global level, it can be any firm or institution, where is important to have real time attendance.

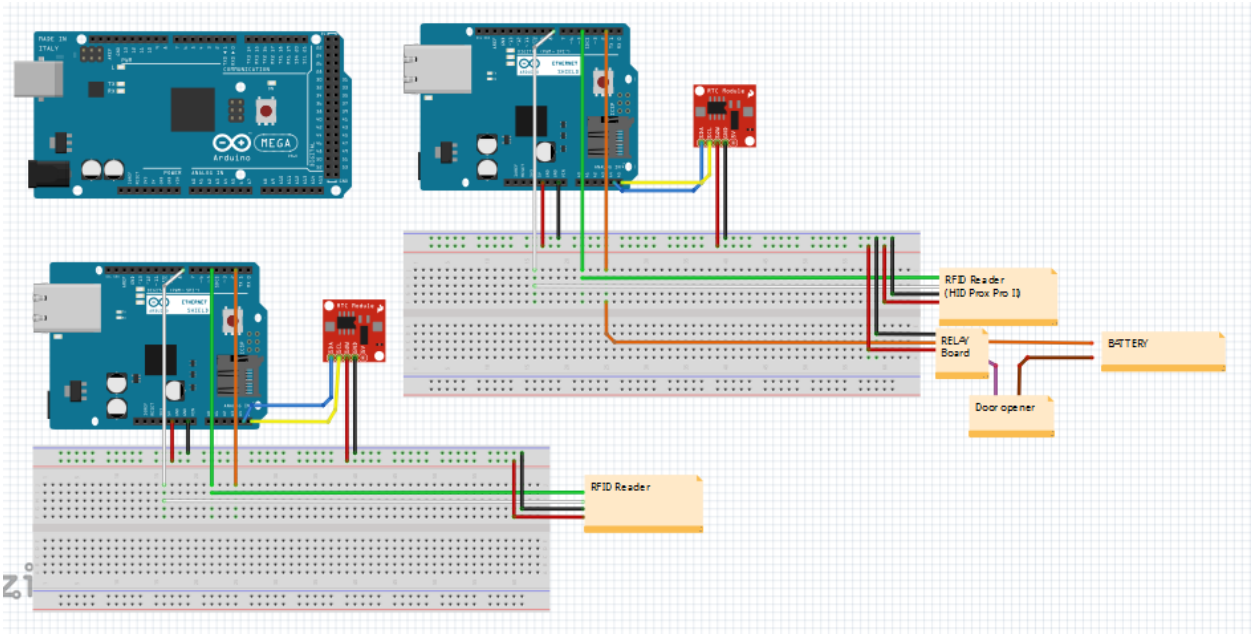


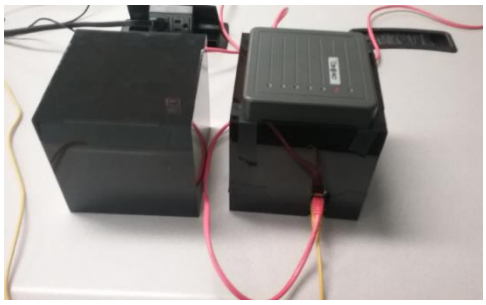
Figure – Schematic of the Access and Attendance System

On the picture above you can see our system. In the left up corner is **Arduino Mega Board**, which with the Ethernet Shield is doing part of the work for the server side. In the left down corner there is Ethernet Shield attached to Arduino and connected to the Real Time Clock (RTC), as well as RFID reader. This is our **attendance system**.

In the right you can see similar system, but in this case except Arduino, Ethernet Shield, RTC and RFID reader, there are additional components such as Relay Board and battery which would be attached to the door. This is working together with Arduino Mega and this represent our **Access System**.

These two systems can be connected to work both in same time, or it can be implemented individually each of them.

How it works?



Attendance system - For example, when student visit the class, everything what need is to tap his or her RFID card on the RFID reader (or box which is special made for this). Information about the students which are attending the class is than sent to the professor on his or her e-mail.

Access System - When person wants to access the premise (room or building), if he or she has permission, that is necessary only to tap his or her RFID card on the reader in from of the door and the door will be open. So, faculty can have information who visited specific premise in the specific time.

Note: In addition, we also implemented this project with Galileo board (instead Arduino Mega) and in this case there is possibility to have all information about access or attendance on the SD card.

Components

Arduino Uno

It is a physical platform for open source computing based on a simple microcontroller board and a development environment for writing the code for the card.

Arduino can be used to develop interactive objects, accepting inputs from a series of sensors or switches, and controlling a variety of lights, motors or other physical outputs. Arduino projects can be independent, or can communicate with software running on your computer (e.g. Flash, Processing, MaxMSP). Circuits can be assembled by hand or purchased preassembled; code-free programming software can be downloaded free.

The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on multimedia programming environment Processing.

Figure 1 – Arduino Uno

Arduino Mega

The difference between Mega and Uno is the port number (and thus available peripherals, for example Mega has more UARTs) and flash memory, RAM and ROM. The processing speed is equal. Almost of the codes that runs in one runs in the other, and often do not even need great care with the ports once the Arduino does this abstraction.

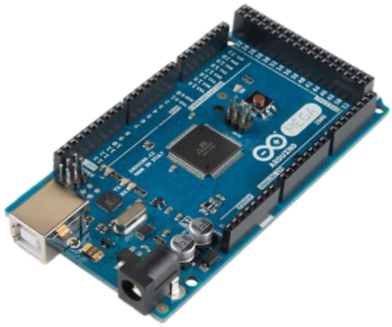


Figure 2 – Arduino Mega

Arduino Ethernet Shield

It provides the necessary interface (circuit) to connect the Arduino to an Ethernet network.

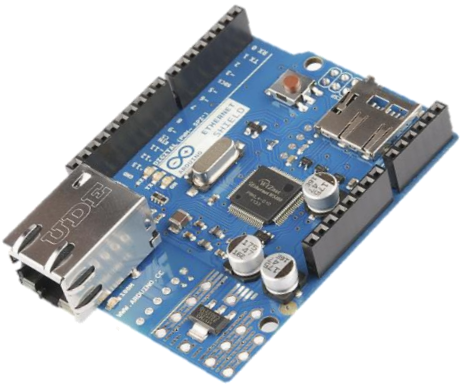


Figure 3 – Arduino Ethernet Shield

RFID Reader

RFID stands for radio-frequency identification. It works transferring data using electromagnetic fields, and has the purpose of electronically identifying stored information in cards or tags. The one used in our project can identify the RFID students' ID cards, reading the necessary information up to ten centimeters away.

Figure 4 – RFID Reader

RFID Card

It contains a chip and an antenna, and is passively powered by the reader/writer when placed in its range. The card also be written and is able to store up to 1 KB of data. It also holds over 100,000 re-writes.

Figure 5 – RFID Card

Real Time Clock

This equipment allows the microcontroller to keep track of time even, if it is rescheduled, which avoids errors or accidents to influence future data collected. The equipment also has a battery backup with an estimated duration of five years.

In our project, its main function is to provide the time, which makes it possible to calculate the student's presence, absence or late arrival in the classroom.

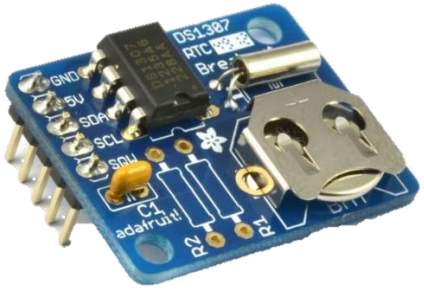


Figure 6 – Real Time Clock

Battery

While choosing the optimum battery for our device, we took into account that it is recommended to use a battery of at least 7V for Arduino Mega to work properly and prevent equipment damages. The closest widely available commercial battery that fulfills this recommendation is a 9V battery, which is the one we decided to use.

Costs

The calculation of the cost of our prototype system was done taking into account the costs of each element we used for both the circuitry and the wood box. All electronic component's prices are available online in http://www.adafruit.com/. Bear on mind that in final system price can be different.

For attendance system:

- *Ethernet Shield - $50.63 (x 2 - one for Arduino Uno, another for Arduino Mega)*
- *Arduino Uno - $24.95*
- *Arduino Mega - $45.95*
- *RFID card reader - $39.95*
- *Real Time Clock board - $9.00*
- *Rest of boards/wires - $5.00*
- *Box - $8.00*
- *TOTAL – $234.11*

For access system would be similar price, with addition of the relay board and the battery.

Advantages

Clear advantages about this project are his multi-possibilities: access and attendance system, sending information in real time via e-mail or simply storing information on SD card.

Previous projects made possibility to store information on SD card and advantage of this project is that now there are possibility that information can be sent directly to the e-mail.

Our solution is effective. We used open source technology such as Arduino, and RFID technology, which is already used in IIT. As you could see price for this project is affordable, not expensive. System is reliable because our tests showed so. There are security reasons why this system should be implemented – many institution needs to have control of access their premises, as well as there is good time not only to save the planet with using less papers, but also to be more effective with sending directly e-mail with attendance list.

Another advantage of this project is that it can be developed more. One of the next steps can be to create online database spreadsheet with attendance, which should not be hard work. Another possibility is to research to make access with voice recognition.

Conclusion

We created this project because we believe that there is the need for this kind of the system. Not only in Illinois Institute of Technology, but this kind of system can be implemented anywhere in the World.

Through this paper you could see our description about embedded system and later concrete system we made. We described you shortly components and parts which our project is consisted of. You could see both schematic for Access and Attendance system.

We wrote about costs based on price of components and we explained advantages, as well as possibilities for farther research or implementation.

We strongly encourage IIT to implement this system, because after all not only University will benefits, but also our students and all employees.

Bibliography

[1] *Programming Arduino Getting Started with Sketches by McGraw-Hill*

[2] *The Arduino Project Book, from Arduino Starter KIT*

[3] *Building Wireless Sensor Networks by Robert Faludi, ISBN-13 – 9780596807733*

[4] *Environmental Monitoring with Arduino, by E. Gertz, ISBN-13 - 9781449310561*

[5] *Arduino official sites: http://www.arduino.cc/ and http://www.arduino.org/*

[6] *ACEbox - RFID-Based Attendance-Checking System, project by Josmar Baruffaldi Cristello, Bianca Canalli Zagueto and Javier Moreno Valdecantos, Illinois Institute of Technology*

[7] *IIT Rice Campus Door System by Duong Alexandre, Illinois Institute of Technology*

[8] Source from this part is: http://www.eetimes.com/document.asp?doc_id=1279478

Image Sources

https://www.honeywellaccess.com/images/HU_5355AGN00_lo.jpg

http://www.ipseitygroup.com/images/sample/sample17old.jpg

http://www.a4papermanufacturers.com/uploads/allimg/140705/1-140F5140H41K.jpg

https://upload.wikimedia.org/wikipedia/commons/3/38/Arduino_Uno_-_R3.jpg

http://www.geek.com/wp-content/uploads/2014/10/11061-01b.jpg

http://site.gravitech.us/Arduino/ARD-ETH/ARD-ETH_1.jpg

http://marelepescar.ro/images/detailed/26/Harti_Romania_si_Bulgaria_Pentru_Sonare_Lowrance.jpg

http://waihung.net/wp-content/uploads/2012/03/IMG_0270.jpg

http://www.iconshock.com/img_vista/IPHONE/mail/jpg/envelope_icon.jpg

http://www.iconsdb.com/icons/preview/soylent-red/door-8-xxl.png

http://www.indiamart.com/altima-security/time-and-attendance-system.html

http://www.netpicks.com/wp-content/uploads/2014/08/profits-150x150.jpg

https://changingjournalism.wordpress.com/2011/05/11/conclusion/

A – Code for Attendance System

```
//Attendance System
#include <SPI.h>
#include <Ethernet.h>
#include <Wire.h>
#include "RTClib.h"
RTC_DS1307 rtc;

char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};
String cardRead = "";
int count =0;
byte mac[] = {
  0x90, 0xA2, 0xDA, 0x0F, 0x4B, 0xD4
};
IPAddress ip(192, 168, 1, 183);
IPAddress server(192, 168, 1, 155);
EthernetClient client;
String dataTosend="";
int classHour=13;
int classMinutes=48;
boolean flaaag=false;

void setup(){
  Serial.begin(9600);
  Ethernet.begin(mac, ip);

 pinMode(6,OUTPUT); //led test
 pinMode(5,INPUT); // data 0
 //pinMode(7,OUTPUT);
 pinMode(8,INPUT); // data 1
  //digitalWrite(7,HIGH);
   if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
  }
```

```
  if (! rtc.isrunning()) {
    Serial.println("RTC is NOT running!");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));

  }

   Serial.println("connecting...");
  if (client.connect(server, 23)) {
    Serial.println("connected");
  }
  else {
    // if you didn't get a connection to the server:
    Serial.println("connection failed");
  }

  delay(3000); // to let the card reader the time to initiate
  Serial.println("system initiated" );
  DateTime now = rtc.now();
}

void loop(){

   int data0 = digitalRead(5);
   int data1 = digitalRead(8);
   int b = -1;

   if(data0==0 || data1==0){

     if(data0==0){
       b=0;
     } else if(data1==0){
       b=1;
     }

     cardRead = cardRead + b;
```

```
     count = count +1;
   }


    if (cardRead.length()==35){
        DateTime now = rtc.now();
        dataTosend="a "+cardRead+" "+now.year()+"/"+now.month()+"/"+now.day()+" "+daysOfTheWeek[now.dayOfTheWeek()]+"
"+now.hour()+":"+now.minute()+"f";
        Serial.println(dataTosend);
        senddata(dataTosend);
        // put the send comand to the server here!!!!!!
        cardRead = "";
        Serial.println("writed in the file!");
        //senddata("s");
  }


  }
void senddata(String s){
client.print(s);
}
boolean getDATE(){
   //Serial.println("entrou na funcao");
DateTime now = rtc.now();
 //Serial.println("passou do now");
  if(now.hour()==classHour && now.minute()==(classMinutes+10)){
   //Serial.println("entrou no dia");
    if(now.dayOfTheWeek()==1 ||now.dayOfTheWeek()==3){
    //Serial.println("entrou na hora");
      return 1;
    Serial.println("email sended");
    //Serial.println("passou do delay");
    }
  }
  else{return 0;}
}
```

Appendix B – Server code (without SD card)

```
//Code for server

#include <SD.h>

#include <SPI.h>

#include <Wire.h>

#include <Ethernet.h>

#define time 1000


//mac

byte mac[] = { 0x90, 0xA2, 0xDA, 0x0F, 0x47, 0x79};

boolean fl=false;

  String Log="";

IPAddress ip(192,168,1, 155);

File myFile;

String stri ="";

String RECORDS="01100111001100000101111010100100011-
PereiraPinto/BrunoAugusto@10100111001100000101000000011110000-
Saldivar/Joseph@00100111001100000110001101001110010-

Zivkovic/Vladimir@00100111001100000101011001111100011-Siron/Craig@";

String AtendanceLOG="";

String TEACHER="11100111001100000101111010100100100"; //joao id


// telnet defaults to port 23

EthernetServer server(23);

//EthernetClient clientPush(80);

boolean alreadyConnected = false; // whether or not the client was connected previously


//email part

byte server2[] = { 200, 147, 99, 132 };; // io do servidor do AOL

EthernetClient client2;


void setup() {

  // initialize the ethernet device

  Ethernet.begin(mac, ip);

  // start listening for clients

  server.begin();
```

```
  //pinMode(53,OUTPUT);
 // Open serial communications and wait for port to open:
  Serial.begin(9600);
  delay(9000);
}

void loop() {
  // wait for a new client:
  EthernetClient client = server.available();

  // when the client sends the first byte, say hello:
  if (client) {
    if (!alreadyConnected) {
      // clead out the input buffer:
      //client.flush();
      Serial.println("We have a new client");

      alreadyConnected = true;
    }

    if (client.available() > 0) {
      // read the bytes incoming from the client:
      char thisChar = client.read();
        stri= stri+thisChar;
    }
    //read until the final of the string
    if(stri.endsWith("f")){
          //seting a String to get the id
          String idreciv="";
          for(int n=2;n<37;n++){
          idreciv=idreciv+stri.charAt(n);

          }
      Serial.println(idreciv);

      //check for attendance system
```

```
 if(stri.startsWith("a")){
   // open the file and look for the name of the student
  Serial.println("entrou no atendence");
  int indexBeg= RECORDS.indexOf(idreciv);
  String Name="";
  if(indexBeg!=-1){
  indexBeg=indexBeg+36;
  int indexEnd=RECORDS.indexOf("@",indexBeg);
  for(indexBeg;indexBeg<indexEnd;indexBeg++){

  Name=Name+RECORDS.charAt(indexBeg);
  }
}
//in the case of the name is not in the record save the id information
else if(idreciv.equals(TEACHER)){
//mandar o email
      Log=AtendanceLOG;
 //code to send email
            fl=true;
          stri="";

}
else if(indexBeg==-1){
    Name=idreciv;
  }

   String date="";
   for(int i=37;i<stri.indexOf("f");i++){
   date=date+stri.charAt(i);
   }
   AtendanceLOG+=Name+" "+date+"\n";
   stri="";

 }
 //check for door lock System
 else if (stri.startsWith("d")){
```

```
            if(RECORDS.indexOf(idreciv)!=-1){
             Serial.println("cardRead is: "+idreciv);
             Serial.println("access granted");
             server.write(2);
             stri="";
            }else if(idreciv.length()==35){

            Serial.println("cardRead is: "+idreciv);
            Serial.println("access negated");
            server.write(1);
            stri="";
            }
      }

    }
    else if(stri.length()==65){
      Serial.println("reading error");
    }
  }

  if(fl==true){
  envia(Log);
  Log="";
  fl=false;
  }

}
void envia(String ddd)
{
Serial.println(ddd);

delay(time);
Serial.println("conecting...");
boolean fla=client2.connect(server2, 587);
```

```
Serial.println(fla);
if (fla)
{
Serial.println("conected!");
Serial.println("sending email...");
Serial.println();
client2.println("EHLO localhost");
recebe();
delay(time);
client2.println("AUTH LOGIN");
recebe();
delay(time);
client2.println("YXJkdWlub2lpdDJAYm9sLmNvbS5icg=="); //  login in 64 base : http://base64-encoder-online.waraxe.us/
recebe();
delay(time);
client2.println("SU5PdGVzdElOTw=="); // Senha in 64 base : http://base64-encoder-online.waraxe.us/
recebe();
delay(time);
client2.println("mail from: <arduinoiit2@bol.com.br>"); //sender
recebe();
delay(time);
client2.println("rcpt to: <bpereira@hawk.iit.edu>"); // reciver
recebe();
delay(time);
client2.println("data");
recebe();
delay(time);
client2.println("Subject: Attendance Log");
recebe();
delay(time);
client2.println(ddd); // scope of the email
recebe();
delay(time);
client2.println("."); // end of email.
recebe();
delay(time);
```

```
client2.println();
recebe();
delay(time);
Serial.println("email sended!");
delay(time);
if (client2.connected()) // Desconect.
{
Serial.println();
Serial.println("desconecting...");
client2.stop();
Serial.println();
Serial.println();
}

}
else
{
Serial.println("connection failed");
}
}

void recebe()
{

while (client2.available())
{
char c = client2.read();
Serial.print(c);
}
```

Appendix C – System implemented with Galileo (instead Arduino Mega)