University of Toulouse

Internship report

Development of methods for the application to Domotic systems

Trainee : Rudy AUBILA

Training : IUT A – Toulouse – GEII Department (Génie Electrique et Informatique Industrielle)

(Electrical Engineering and Industrial Computing)

Company : CELTAB (CEntro Latino-americano de Tecnologias ABertas)

(Latin American Open Technology Center)

Company tutor : David JOURDAIN

Academic tutor : Luiz Fernando LAVADO VILLA

Period : from 06/04/2015 to 26/06/2015

Thanking

I would like to thank Mr. Luiz Fernando LAVADO VILLA who initiated the project, Ms. Stephanie COMBETTES who helped in the organization and administrative procedures of the trip and my parent to their financial help.

My thanks also go to Mr. David JOURDAIN who supervised and mentored my work, for his attention, and for all the time he dedicated to me, his advice was really useful and appreciated.

I also thank all the member of the CELTAB with who I worked, who were always ready to help me, and more particularly Mr. Arthur GARCETE, Mr. Kenner KLIEMANN and Mr. Augusto DANTAS for their patience and the time they spent to explain their work to me.

Finally, I would like thank Ms. Adiles POMAGERSKI who housed and fed me during my stay, Gustavo VALIATI helped me to integrate myself in Brazil, and Thiago BITENCOURT for taking me to work by car everyday.

Summery

Introduction

To complete my training of Electrical Engineering and Industrial Computing, I did a twelve-week internship. As my colleagues at IUT, I started my searches at the beginning of this year, but I didn’t have any positive answer. I met Mr. Luiz Fernando LAVADO VILLA who comes from Brazil and who had just arrived at the IUT, and we spoke about the possibility of doing an internship in Brazil. I had confirmation quickly and I began to prepare this trip.

During my first year at the IUT (2012-2013), I assisted a presentation on the internship in Quebec. I was very interested then but I redoubled my first yearwhich changed my plans. I wanted to make my second year in apprenticeship, but I did not find a company. I followed the classic training, disappointed. A few months later, my desire to do an internship abroad returned when Luiz spoke me about Brazil. So I did my internship in Foz do Iguaçu, Parana, Brazil, on the others side of the world, in a different culture, away from all everything I knew.

A lot of things happened during these three months, I will divide this report into three parts: I will speak about the Company who employed me, then I will make a presentation of work that I realized. That will be the most important part of this report. Finally, I will analyze this internship experience as a whole.

I) The Company

The CELTAB (Latin American Center for open technologies) was created in 2013 with plans to become a center for Research & Development of innovative solutions using only free technologies. Thereafter, these solutions will be shared with the rest of Latin America. CELTAB belongs to the Itaipu Technological Park (PTI) located in the three-border region (Brazil, Argentina and Paraguay) in Foz do Iguaçu for the Brazilian part where I worked.

A few words about the PTI. Founded in 2003 by Itaipu Binational (the largest hydroelectric plant in the world in the production of energy), the park extends from about 116 hectares and will be extended by additional 40 hectares. It’s a small town on its own, with restaurants, grocery store, banks... This structure promotes education Research & Development of science, technology and entrepreneurship. Therefore, the PTI is partnering with four universities: the State University of the West of Paraná (Unioeste), the Open University of Brazil (UAB), the Technical School of Brazil (E-Tec) and the Federal University of Latin American Integration (Unila), about 2,000 students (including 200 on the park) and a research laboratory as CELTAB, which represents about 2,000 employees including the service maintenance of the park.

The CELTAB is the result of an agreement between Itaipu and PTI. It is an applied research unit in the PTI. In practice, it responds to the problem related to technology that meets the company but also develops solutions to to implement these answers. For researchers who work there, it's a way to do whatever they please to them (in computer field), while obtaining results. Indeed, due to the use of free software, the team of researchers of CELTAB can develop their knowledge of absolutely all possible applications in the world of computing. Any difficulties encountered during the implementation of a project, are dissipated by sharing ideas and solutions within the team, for faster results.

II) Work done

1) The beginning of the internship

For my first working day, each member of the CELTAB with who I would work, introduced me to their different projects. David decided that the most interesting for me should work on the “Domotic Project” with Kenner and Augusto. But before, I needed to improve my knowledge on how to use Linux, a protoboard, Arduino, nRFL01 and atmega328p micro-controllers (see appendix n°3 for details).

Also I had to learn Portuguese to know how to ask for help, and how to explain a problem. It was difficult but I learned fast the base of this fascinating language.

I was trained to use Linux by Arthur during three sessions of two hours to practice the Linux’s commands. Then, Kenner assigned me to a task. I had to realize three mini projects to familiarize myself with the protoboard. I will present you this task below.

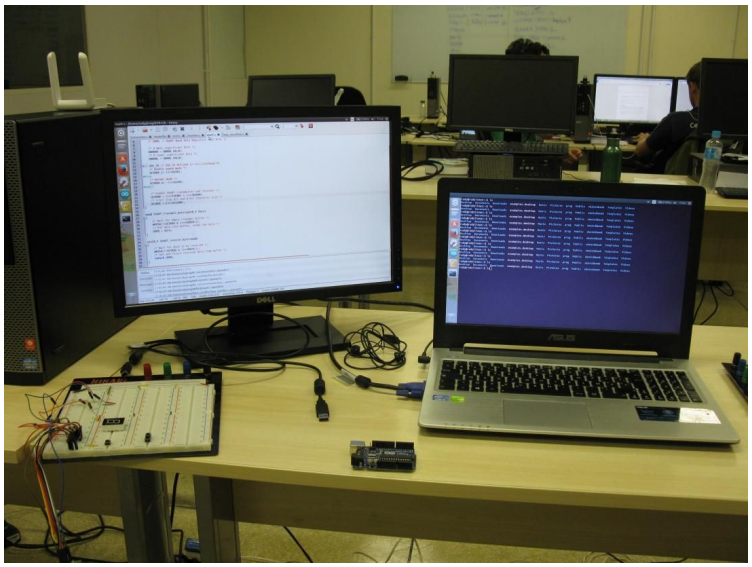


Figure 1 – My workspace

2) Steps

i. Task #1 – AVR-Training

Kenner introduced me the website github.com. It is a resource sharing platform. He sent me a list of missions to do before joining “Domotic project”:

a. Build your first hello world with AVR

b. Build a single way traffic light

c. Learn about Free and Open Source Software LICENSES

d. Improve the traffic light to two-way, with time synchronization

e. Build a led manager (DIM LEDS) with USART communication using USB2SERIAL

f. Improve the synchronization to RF-communication, using NRF24L01

I will present you all my work in this part, and the analyze and the evolution of my knowledge in the next part

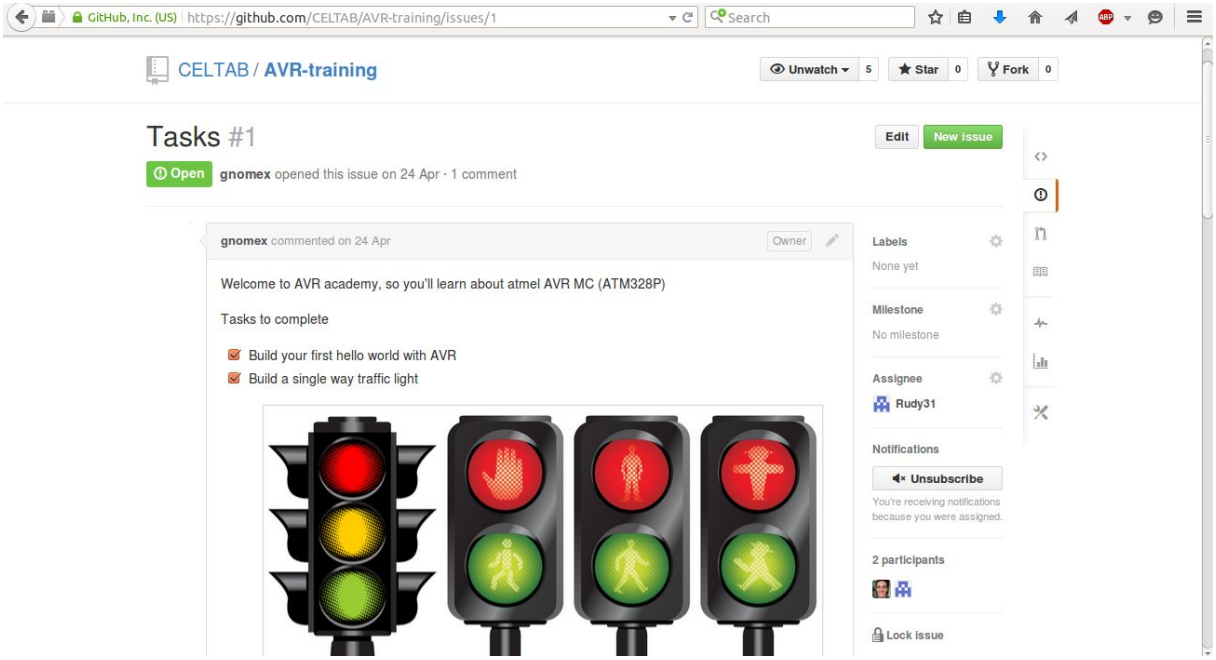


Figure 2 – My Github issues

ii. Hello word

In the computing world, to build a “Hello word” means to write the minimum to see this sentences when the code is executed. It’s the simplest program.

I had for mission to build my first “Hello world”, that means I had to make a simple program. I started by to blink a LED, to see the use of protoboard. I connected my Arduino with the micro-controller, I connected a LED and a resistor. Then I configured the register corresponding to the pin connected to the LED in output. Finally, I ordered the sending of a high and a low state separated by a period to see the LED blink.

iii. Single way traffic light

For my second mission, I used three LED to simulate traffic lights. It was as simple as the “Hello world”. I just added two LED, modified the register to that is takes into consideration the additional outputs, and I built a sequence to make it look like a traffic light. After that, I started my first project by repeating what I had done for this test and developing it.

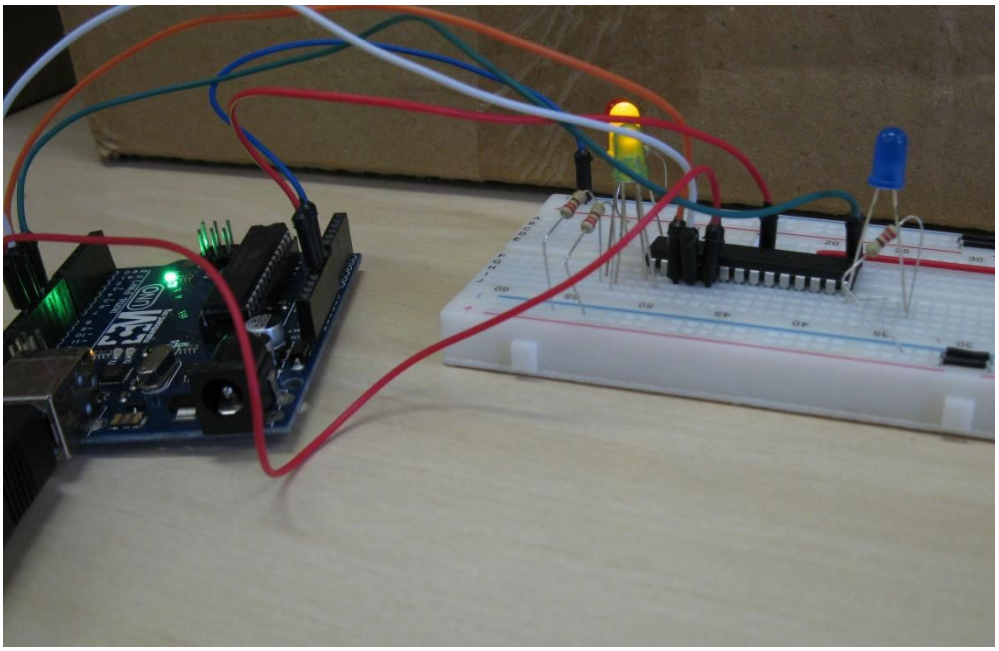


Figure 3 – First test for my traffic light

iv. Open Source Software

The company is turned on the share and the open source. I had to read and to understand this point of view. This is what I have withheld:

To share your work is to allow collaborations to develop and correct the program. Everyone can modify and improve your code. Even if they are small projects, and it is unlikely that someone develops it for a large-scale industrial uses, now my code is on this platform, other programmers could look over, fix bugs and add other options.

v. Two-way traffic light

I made a program that simulates a cross with two streets and a press button for pedestrian's call. (see appendix n°1)

The green LED switch on, wait and switch off, then the orange do the same thing and the red too. Afterwardsit is the turn of the second street and so on. There is one blue LED on each street representing if the pedestrian can across the street or not. If on one street the green LED is switched on, then on the second streed it's the red one and the blue one that turn on. Before changing the traffic lights, the blue LED blinks to warn that it the end of time to cross.

I wanted to use an interrupt function, because we learned to use in "Study and Realization" at IUT. I added the option of the press button. The interruption stops the current action to do something else. When the user press the button, the red LED comes more quickly, for the pedestrian can cross faster, and the other street keep a normal running.

I didn't meet a lot of problems in this task because I already realized that kind of project at the IUT. The most difficult was to find the good sequence in the main function, to have not a jump between two wrong steps.

I used Geany, an editor software to write my code. The novelty was the procedure to execute my code, I had to use the terminal of Linux to compile and flash my program.

Finally, when the program was running well, I sent it on github to save and added a "read-me" to give information to an eventual next user.

vi. USB2Serial : 7 segments display

For the next missions, I had to build a LED manager. I chose to take a 7 segments display and I created a program that writes on this device a number between 0 and 9 and a letter between A and F (see appendix n°2)

To communicate with the display, I used another device, the USB2Serial (see appendix n°3) and the software CuteCom to exchange data with the micro-controller.

I took the initiative to build two different functions: the first one for when the user tap on keyboard an existing character. This character is then written on the display. The second function is an hexadecimal counter (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 0, ...).

Once the program was finished, I chose to add some options and to upgrade the user interface. I added a press button to start an interruption that makes a countdown. I also added some sentences so that the user knows what the microcontroller visualizes.

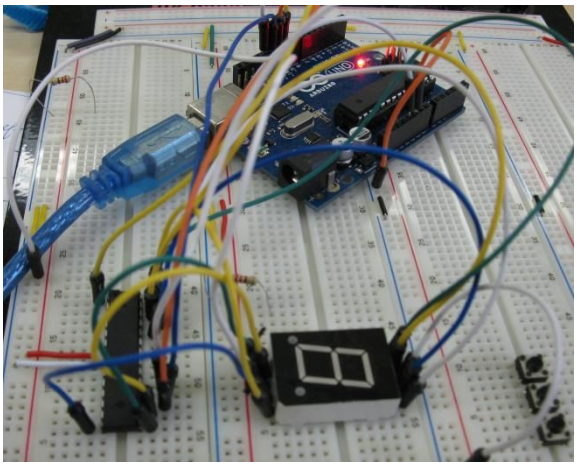


Figure 4 – 7 segment display

vii. RF-communication :

For the last mission of this first task, I had to use the component nRFL01 to build a RF-communication. The idea here was: on one side, to have an Arduino and this device, send data (characters string) to another side with the same device who receive and write the data on CuteCom. But I met some difficulties to realize this project. I passed some days, I tried to change Arduino, the nRFL01, but without success.

My second plan was to find an example on internet to test it. I found two examples of the same kind of project. I had a press button on one side and a LED on the other side. The theory was, when the button is pressed, the LED lights on. But for the two examples, I had problems of libraries, and even with the help of my colleagues, I have not found a solution.

Finally, Augusto worked on library to permit me to programming in C. So, I tried to realize a project, but I was already the end of the internship.

III) Results

Connection to the university

During this internship, I could practice several skills I have acquired at IUT and developed new ones.

I progressed in English because I had no choice, to communicate with people at work and outside too. I learned a little Portuguese to be able to talk with people who don’t speak English like Adiles, and to try understand the amusing reflections.

The main part of my internship it was to think about a problem and to program to achieve it. To succeed, I used the same approach than in “Study and Realization”. I had to read the datasheet of the components that I used, to watch the existing function to be able to use it. Programming to find how to solve my problem and executing my code to test it.

Then, with the training I have received on Linux commands, I was able to improve the knowledge we have not had time to develop in “Network”. It’s the same thing for the “Bus of communication”, I improved my knowledge about Arduino and object oriented programming, with the RF-communication project.

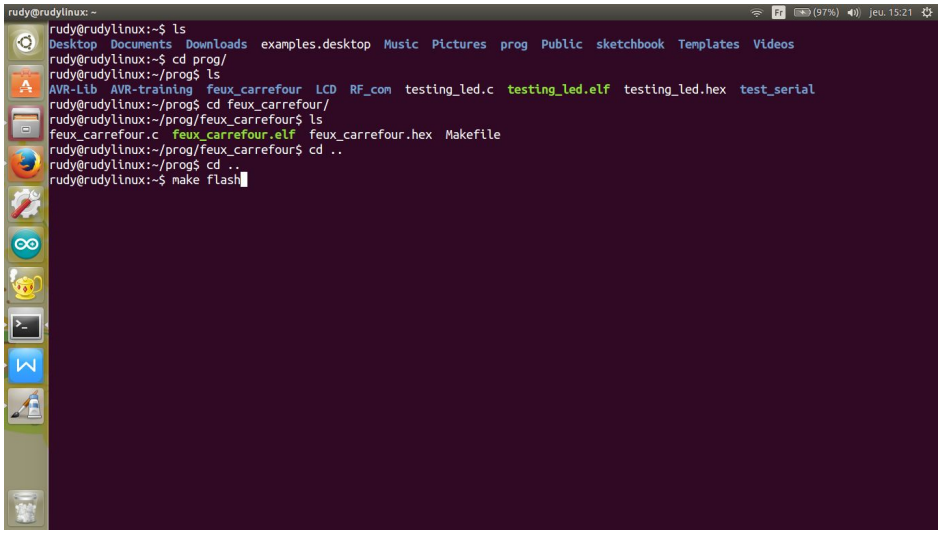
Difficulties faced

During my internship, in a new and beautiful place, I learned a lot but I had to face some difficulties:

In a first time, the language was a problem, because it is the Portuguese is spoken in Brazil, and I didn’t speak it. Thankfully, in Celtab, all people speak English. But I had the same problem because I didn’t speak very well in English and it is difficult to think for the technical work and stay concentrate to explain or to ask something in English. I used a lot the translator for the beginning, and gradually, I spoke better.

Figure 5 – Language difficulties

Then, on my first day, they asked me if I preferred to work on Window or Linux. I answered Window because I used that since I used PC, but it was not the answer they expected. I had to accept to work on Linux and Arthur taught me about it. After few days, I installed Linux Ubuntu on my own laptop to bring back my work at home for more convenience. Now I use the terminal of Linux to work and if I forget command, I ask someone or I consult the forum dedicated to Linux.



The last difficulty I had was using the Arduino and objects oriented programming, I had to adapt myself but I favored programming in C. Indeed, instead of using the Arduino IDE software to make my code, I configured it to use the micro controller. With the datasheet of the micro controller, I could connect it with the Arduino. After that, I just had to write my code on the editor software Geany, and flash it with the terminal.

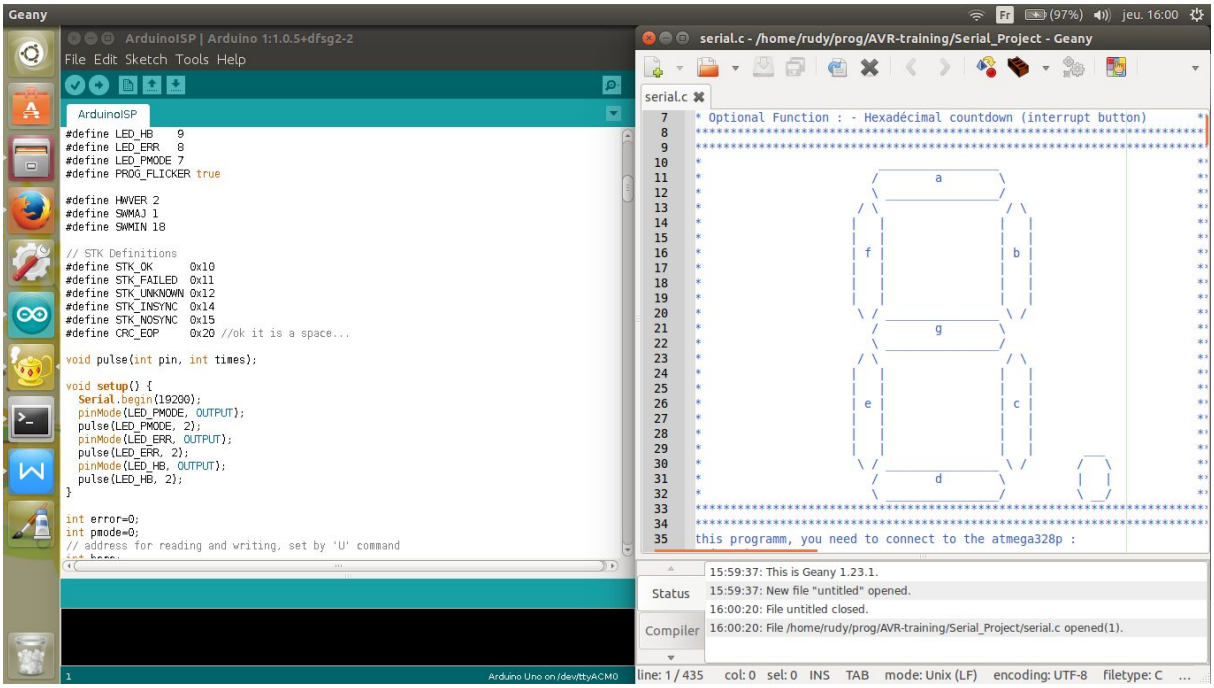


Figure 1 – Passage from oriented to no oriented programming

Professional project

I chose to follow the electrical engineering formation, because I like the field of the electronic and the industry. I had a professional project when I arrived at IUT, but this internship has completely changed my plans. I will explain how in two points.

First of all, I had a scientific training before IUT. I redoubled my first year and I knew I did not have the level for an engineer school. My plan was to find work after IUT or maybe make a license. But what I especially wanted was to work in maintenance or production. To manipulate and repair motivated me. But during this internship, I was behind a desk, I did programming and I executed codes on little simulation (like the traffic light...), and I liked it. That looked like practice of “Study and realization”, but with more time and no other courses around. Now, I want to find a work and be in this same situation.

Then, I found that doing my internship abroad was an incredible experience. If the opportunity arises again, I would not hesitate. To travel and to meet people with a different culture is very rewarding. It seems difficult at first, but the work is the same, just need to adapt to be able to interact with others.

Conclusion

This internship was a great and unexpected experience. Indeed, just one year ago I would have never believed coming to Brazil, so long and to work here. It is an experience that I wish to everyone. Professionally, I have gained new knowledge and perfected the knowledge that I have received at IUT. It is true that I was not on major projects with stakes for the company and considerable constraints to be taken into account, but this experience allowed me to discover the professional environment related to my training.

This internship also allowed me to change my point of view about Study and Research & Development. I had plans to find a job in the production or maintenance, more manual than cerebral craft. But with this experience, and a summer job in a supermarket that was not done for me, I could see that trades oriented to the study suit me better.

Web List

Picture of atmega328p

http://www.vwlowen.co.uk/arduino/bootloader/pinmap.jpg

Picture of nRF24L01

http://cloud1.lbox.me/images/384x384/201206/electronics-diy-nrf24l01-2-4ghz-wireless-communication-module_yfkgks1339666554579.jpg

Picture of language problem

http://thumbs.dreamstime.com/x/communication-problem-17452102.jpg

Logo of Celtab

http://www.celtab.org.br/

Logo of PTI

http://www.pti.org.br/

About the company

http://2014.latinoware.org/centro-latino-americano-de-tecnologias-abertas-sera-apresentado-na-latinoware/

https://www.itaipu.gov.br/en/technology/itaipu-technological-park-itp-0

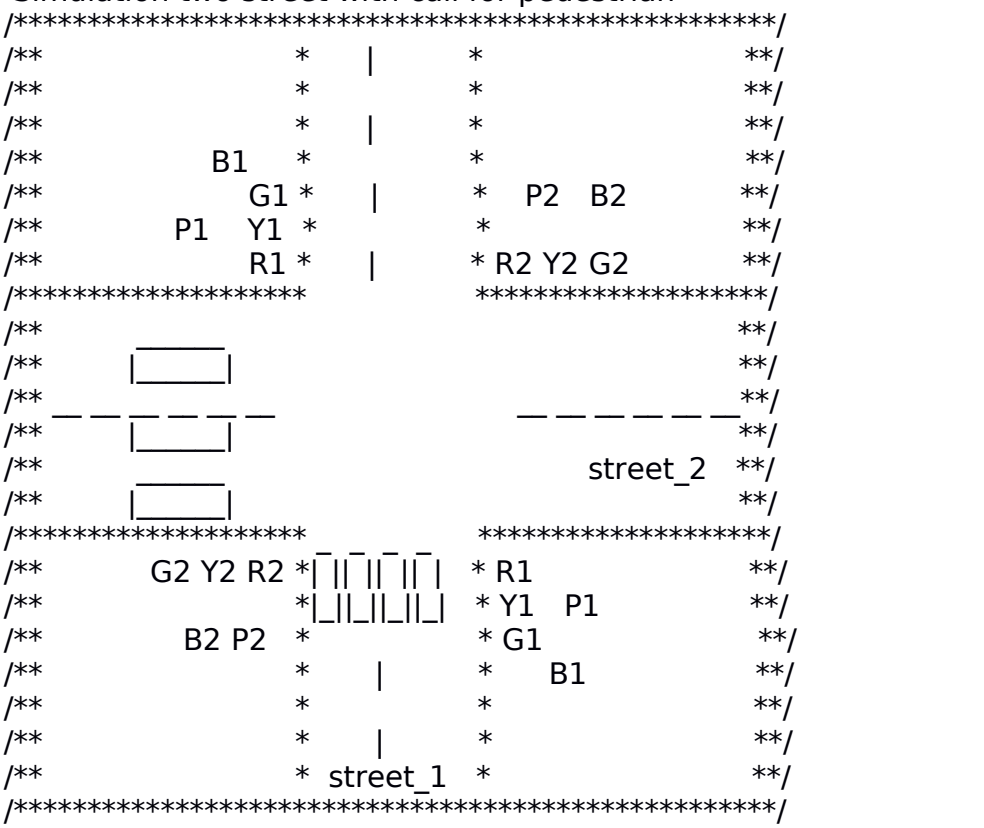
http://www.pti.org.br/en

Appendix

Summery

Appendix n°1 : Traffic lights

Program for Traffic Lights
Simulation two street with call for pedestrian




```
/** To use this program, you need to connect to the atmega328p :
/**
/** Red_led_1       on PB0        (street_1)  :  R1
/** Red_led_2       on PC2        (street_2)  :  R2
/**
/** Yellow_led_1    on PB1        (street_1)  :  Y1
/** Yellow_led_2    on PC1        (street_2)  :  Y2
/**
/** Green_led_1     on PB2        (street_1)  :  G1
/** Green_led_2     on PC0        (street_2)  :  G2
/**
/** Pedestrian_led_1 on PC5     (street_1)  :  P1
/** Pedestrian_led_2 on PC3     (street_2)  :  P2
/**
/** Button_1        on PD2        (street_1)  :  B1
/** Button_2        on PD3        (street_2)  :  B2
/**
/** For the connection with the Arduino UNO :
/**  ~11           on PB3
/**  12            on PB4
/**  13            on PB5
/**  ~10           on PC6 (Reset)
/**  5v            on Vcc
/**  Gnd           on Gnd
/*****************************************************/
```

```
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

unsigned int state = 0;

void blink_blue_1 (void){           /*********************************/
        PORTC |= (1<<PC5)           /** Blue_led_1 blinks            **/
        _delay_ms(200);             /*********************************/
        PORTC &= ~(1<<PC5);
        _delay_ms(200);
}
void blink_blue_2 (void){           /*********************************/
        PORTC |= (1<<PC3);          /** Blue_led_2 blinks            **/
        _delay_ms(200);             /*********************************/
        PORTC &= ~(1<<PC3);
        _delay_ms(200);
}
void green_1 (void){                /*********************************/
        PORTB = 4;                  /** Green_led_1 is turned on  **/
        _delay_ms(3000);            /*********************************/
}
void green_2 (void){                /*********************************/
        PORTC |= 1<<PC0;            /** Green_led_2 is turned on  **/
        _delay_ms(3000);            /*********************************/
}
void orange_1 (void){               /*********************************/
        PORTB = 2;                  /** Yellow_led_1 is turned on **/
        _delay_ms(1500);            /*********************************/
}
void orange_2 (void){               /*********************************/
        PORTC |= 1<< PC1;           /** Yellow_led_2 is turned on **/
        _delay_ms(1500);            /*********************************/
}
void red_1 (void){
        PORTB = 1;                  /** Red_led_1 is turned on     **/
}
void red_2 (void){
        PORTC = 4;                  /** Red_led_2 is turned on    **/
}
void blue_1 (void){
        PORTC = 32;                 /** Blue_led_1 is turned on   **/
}
void blue_2 (void){
        PORTC |= 1 << 3;            /** Blue_led_2 is turned on   **/
}
```

```
void step_1 (void){
	red_2();
	blue_2();
	green_1();
}
void step_2 (void){
	red_2();
	blue_2();
	orange_1();
}
void step_3 (void){
	int i;
	red_2();
	red_1();
	for(i = 0; i<10; i++){
		blink_blue_2();
	}
}
void step_4 (void){
	red_1();
	blue_1();
	green_2();
}
void step_5 (void){
	red_1();
	blue_1();
	orange_2();
}
void step_6 (void){
	int i;
	red_1();
	red_2();
	for(i = 0; i<10; i++){
		blink_blue_1();
	}
}
void walk_call_1 (void){
	if ((PORTB == 4)&(state != 2)){		/*********************************/
		step_2();				/** Red_led_1 comes faster    **/
		step_3();				/*********************************/
		state = 1;
	}
}
void walk_call_2 (void){
	if ((PORTC == 33)&(state !=1)){		/*********************************/
		step_5();				/** Red_led_2 comes faster   **/
		step_6();				/*********************************/
		state = 2;
	}
}
ISR(INT0_vect){
	walk_call_1();		/** interrupt when the button_1 is pressed **/
}
ISR(INT1_vect){
```

```
        walk_call_2();        /** interrupt when the button_2 is pressed **/
}
int main (void){
        DDRB = 7;                   /** Put PB0, PB1 and PB2 in Output**/
        DDRC = 47;                  /** Put PC0, PC1, PC2, PC3 and PC5 in Output**/

        DDRD = 1<<PD2;                          /*********************************/
        DDRD |= 1<<PD3;                         /*********************************/
        PORTD = 1<<PD2;                         /**                             **/
        PORTD |= 1<<PD3;                        /**                             **/
        EIMSK = 1<<INT0;                        /**  Required to use interrupt  **/
        EIMSK |= 1<<INT1;                       /**                             **/
        MCUCR = 1<<ISC01 | 1<<ISC00;            /**                             **/
        MCUCR |= 1<<ISC11 | 1<<ISC10;           /*********************************/
        sei();                                  /*********************************/
while(1){
        if (state != 2)        {
                state = 0;
                step_1();
        }
        if (state == 1){
                step_4();
                step_5();
                step_6();
        }
        else    {
                if(state != 2){
                        step_2();
                        step_3();
                }
        }
        if (state != 1)        {
                state = 0;
                step_4();

        }
        if (state == 2){
                step_1();
                step_2();
                step_3();
        }
        else{
                if(state !=1){
                        step_5();
                        step_6();
                }
        }
}
return 0;
}
```

Appendix n°2 : 7 segments display

```
/*****************************************************************************/
/****                    7 segments display (YSD-160AR4B-8)                 **********/
/**** 2 Functions : - Displays the character written on the keyboard   **********/
/****                  - Hexadecimal counter                                **********/
/*****************************************************************************/
/**** Optional Function : - Hexadécimal countdown (interrupt button) **********/
/*****************************************************************************/
/****                                                                       **********/
/****             _________________                                         **********/
/****            /         a       \                                        **********/
/****            \ _________________/                                       **********/
/****           / \                 / \                                     **********/
/****          |   |               |   |                                    **********/
/****          |   |               |   |                                    **********/
/****          | f |               | b |                                    **********/
/****          |   |               |   |                                    **********/
/****          |   |               |   |                                    **********/
/****          |   |               |   |                                    **********/
/****           \ / _______________ \ /                                     **********/
/****            /         g       \                                        **********/
/****            \ _________________/                                       **********/
/****           / \                 / \                                     **********/
/****          |   |               |   |                                    **********/
/****          |   |               |   |                                    **********/
/*****          | e |               | c |                                    **********/
/*****          |   |               |   |                                    **********/
/*****          |   |               |   |                                    **********/
/*****          |   |               |   |     ____                           **********/
/*****           \ / _______________ \ /     /    \                         **********/
/*****            /         d       \       |    |                         **********/
/*****            \ _________________/       \ ___ /                         **********/
/*****************************************************************************/
/*****************************************************************************/
/** To use this programm, you need to connect to the atmega328p :
/** Segments (PINS) :
/**   - a (7)    : PB0        - e (1) : PC1
/**   - b (6)    : PB1        - f (9) : PC2   PINS 3 & 8 connected to Vcc
/**   - c (4)    : PB2        - g (10): PC3
/**   - d (2)    : PC0        - DP(5) : PC4
/**   - Button_1  : PD2
/**   - TXD      : PD0(RXD)    - RXD   : PD1(TXD)   (USB2Serial)
/**
/** For the connection with the Arduino Uno :
/**  ~11           on PB3
/**  12            on PB4
/**  13            on PB5
/**  ~10           on PC6 (Reset)
/**  5v            on Vcc
/**  Gnd           on Gnd
/*****************************************************************************/
```

```
/** Library **/
#include <usart.h>
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <avr/pgmspace.h>
//#include <stdio.h>

/** Declaration for CuteCom Interface **/
uint8_t state;
tab_choose     [60] = "\n function 0: enter character  1: hexadecimal counter\n";
char tab_letter [26] = "\nYou just typed a letter : ";
char tab_number     [26] = "\nYou just typed a number : ";
char tab_exit       [32] = "\nExit function : enter character";
char tab_timec      [59] = "\nHow many time would you like counting between 0
and 9 ? : ";
char tab_interrupt  [26] = "\nYou just pressed a button ";
char tab_wrong      [13] = "\nWrong value ";
char tab_count      [26] = "\nThe program has counted : ";

/** Decoding 7 segments **/

void write_0 (void){
        PORTB = 0 ;
        PORTC = 24;
}
void write_1 (void){
        PORTB = 1 ;
        PORTC = 31;
}
void write_2 (void){
        PORTB = 4 ;
        PORTC = 20;
}
void write_3 (void){
        PORTB = 0 ;
        PORTC = 22;
}
void write_4 (void){
        PORTB = 1 ;
        PORTC = 19;
}
void write_5 (void){
        PORTB = 2 ;
        PORTC = 18;
}
void write_6 (void){
        PORTB = 2 ;
        PORTC = 16;
}
void write_7 (void){
        PORTB = 0 ;
        PORTC = 31;
}
```

```
void write_8 (void){
        PORTB = 0 ;
        PORTC = 16;
}
void write_9 (void){
        PORTB = 0 ;
        PORTC = 18;
}
void write_A (void){
        PORTB = 0 ;
        PORTC = 17;
}
void write_B (void){
        PORTB = 3 ;
        PORTC = 16;
}
void write_C (void){
        PORTB = 6 ;
        PORTC = 24;
}
void write_D (void){
        PORTB = 1 ;
        PORTC = 20;
}
void write_E (void){
        PORTB = 6 ;
        PORTC = 16;
}
void write_F (void){
        PORTB = 6 ;
        PORTC = 17;
}
void write_DP (void){
        PORTB = 7 ;
        PORTC = 15;
}
char write_car (char chx){
switch (chx){
        case '0'   : write_0();
        break;
        case '1'   : write_1();
        break;
        case '2'   : write_2();
        break;
        case '3'   : write_3();
        break;
        case '4'   : write_4();
        break;
        case '5'   : write_5();
        break;
        case '6'   : write_6();
        break;
        case '7'   : write_7();
        break;
```

```
        case '8'   : write_8();
        break;
        case '9'   : write_9();
        break;
        case 'a' : write_A();
        break;
        case 'b' : write_B();
        break;
        case 'c' : write_C();
        break;
        case 'd' : write_D();
        break;
        case 'e' : write_E();
        break;
        case 'f' : write_F();
        break;
        case '.' : write_DP();
        break;
        case '@' :
        break;
        }
return chx;
}

void counter (void){
        state = 0;
        char i  = 0x30;
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
```

```
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i = 0x61;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i++;
        }
        if (state == 0){
                write_car(i);
                _delay_ms(250);
                i = 0x30;
        }
}
```

```
void write_tab (char *tab, uint8_t size){
	uint8_t cpt;
	for (cpt = 0; cpt<size; cpt++){
		USART_transmit_byte(tab[cpt]);
	}
}

/** Interruption **/

ISR(INT0_vect){
	char i;
	//uint8_t state;
	write_tab(tab_interrupt, 26);


	for(i = 0x66; i>=0x61; i=i-1){
		write_car(i);
		_delay_ms(250);
	}
	for(i = 0x39; i>=0x30; i=i-1){
		write_car(i);
		_delay_ms(250);
	}
	//write_car('0');
	state = 1;
}

int main (void){

DDRD = 1<<PD2;                        /*********************************/
PORTD = 1<<PD2;                       /*********************************/
EIMSK = 1<<INT0;                      /** Required to use interrupt  **/
MCUCR = 1<<ISC01 | 1<<ISC00;  /*********************************/
sei();                                /*********************************/

DDRB = 7;          /*************************************************/
DDRC = 31;         /** PB0, PB1 and PB2 output config              **/
PORTB = 7;         /** PC0, PC1, PC2, PC3 and PC4 output config  **/
PORTC = 31;        /*************************************************/

char choose;
char chx;
uint8_t state =0;
USART_init();
char count_time = 0x30;
char cpt_2;
state = 0;
```

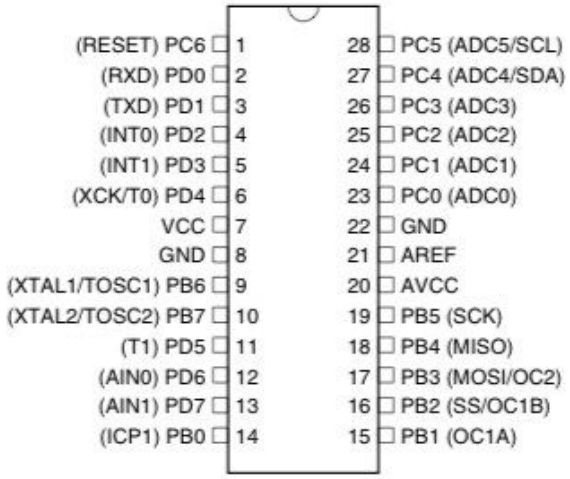
```
while(1){
	write_tab(tab_choose, 60);
	choose = USART_receive_byte();
	USART_transmit_byte(choose);
	switch (choose){
		case '0' :	do{
					chx = USART_receive_byte();
					chx = write_car(chx);
					if ((chx >= 0x30)&&(chx <= 0x39)){
						write_tab(tab_number,26);
					}
					if ((chx >= 0x61)&&(chx <= 0x66)){
						write_tab(tab_letter,26);
					}
					USART_transmit_byte(chx);
				}while (chx != 0x40);
				write_tab(tab_exit,32);

		break;

		case '1' :
				cpt_2 = 0x30;
				write_tab(tab_timec,59);
				count_time = USART_receive_byte();

				USART_transmit_byte(count_time);

				if ((count_time < 0x30)||(count_time > 0x39)){
					write_tab(tab_wrong, 13);

				}
				else	{
					do{
						counter();
						cpt_2++;
						write_tab(tab_count, 26);
						USART_transmit_byte(cpt_2);
					}
					while (cpt_2 < count_time);
				}
		break;

		default  :
		break;
	}
	state = 0;
}
return 0;
}
```

Appendix n°3 : Devices and Components

I) Atmega328p

Here is the micro-controller atmega328p:

This picture indicate the function of each pin, to help for the connections with the Protoboard. For example, I used PB0, PB1, PB2 to link the LED, PD2 to link the press-button for interruption. PB3, PB4, VCC, GND were connected to the Arduino.



II) Arduino UNO

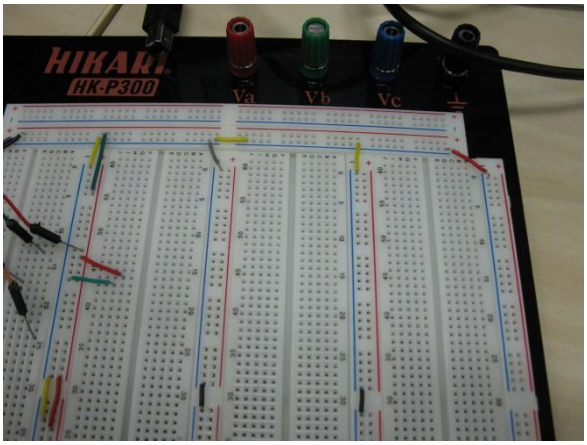
Here is Arduino :

It is the same that I already used at IUT. After the connections with the micro controller, the card need to be connect with the PC by USB port.

III) Protoboard

Here is a protoboard:

It is a device for connecting components, you must do the connections between the different lines to make electric circuits. I have used it for connect micro-controller with the Arduino and LED. The blue line corresponding at GND and the red line is Vcc (5v).



Here is a nRF24l01 :

It is a 2,4GHz Transceiver. With this component, the Arduino and the micro-controller, I saw the the features of a RF communication, each pin have function to communicate with the computer by the Arduino or the micro-controller without to lose data.