



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Sergio Javierre

HARDWARE DESIGN FOR A SMART
LOCK SYSTEM FOR HOME AUTOMA-
TION

School of Technology
2016

ABSTRACT

Author	Sergio Javierre
Title	Hardware Design for a Smart Lock System for Home Automation
Year	2016
Language	English
Pages	49
Name of Supervisor	Smail Menani

Developing a system that can be controlled by a portable device and easily implemented on any door is the main goal of the Smart Lock System. Its purpose is to avoid the usage of a hardware key; the new key will be an Android app in the mobile device which provides security to the user and to the specific area due to the fact that only restricted personnel is permitted access in this area.

The design of the embedded system and its implementation, focusing on the system hardware part, are the key aims of this thesis. PADS-PCB design software is used in order to develop and manufacture the PCB board. All the material and equipments to manufacture the prototype are provided by the Technobothnia laboratory.

The whole system was made available into a door prototype to make realistic tests. In this test phase, the Smart Lock System successfully controlled the state of the electric strike; the communication between the mobile phone and the Bluetooth Module worked successfully and the data was checked and passed correctly. Hence, the goals of the project are achieved and a reliable lock system is developed.

CONTENTS

ABSTRACT

LIST OF FIGURES AND TABLES

1	INTRODUCTION	6
2	SMART LOCK SYSTEM OVERVIEW	8
2.1	Hardware Structure	9
3	EMBEDDED SYSTEM DESIGN	12
3.1	Atmega32 Microcontroller.....	12
3.2	HC-05 Bluetooth Module	14
3.3	Door Strike.....	16
3.4	Power Supply Circuit.....	21
3.5	Indoor Push Button	23
3.6	SPI Port.....	24
3.7	Other parts.....	25
3.7.1	Oscillator	26
3.7.2	LED's	26
3.7.3	Reset Button	27
3.7.4	Capacitor in pin 10.....	28
4	EMBEDDED SYSTEM IMPLEMENTATION	29
4.1	PADS PCB Design Software	29
4.1.1	PADS Logic	29
4.1.2	PADS Layout	31
4.2	PCB Manufacturing	33
4.2.1	Provisioning	33
4.2.2	Vacuum UV Exposing Device	34
4.2.3	PCB Manufacturing Machine	35
4.2.4	Drilling machine.....	37
4.2.5	Soldering-iron	37
5	SYSTEM TESTS AND RESULTS.....	39
6	CONCLUSIONS AND IMPROVEMENTS.....	45
	REFERENCES	

LIST OF FIGURES AND TABLES

Figure 1: System overall structure.	8
Figure 2: Schematic of the embedded system for the Smart Lock.	10
Figure 3: Schematic of the electronics of the door lock system.....	12
Figure 4: Pinout Atmega32.	13
Figure 5: HC – 05 Bluetooth Module.	14
Figure 6: Electric door strike used in this project	17
Figure 7: Door strike circuit.	20
Figure 8: Power supply circuit.	21
Figure 9: DC barrel power jack.....	22
Figure 10: Circuit for the indoor push button.	23
Figure 11: Connecting SPI Port.	25
Figure 12: AVRISP mkII.	25
Figure 13: Crystal Oscillator Connections.....	26
Figure 14: LED’s used to show the system state.	26
Figure 15: Reset button design.....	27
Figure 16: Stabilizer capacitor	28
Figure 17: Result of PADS Logic design.....	30
Figure 18: Result of PADS Layout.	32
Table 1: Material List	34
Figure 19: Vacuum UV Exposing Device.....	35
Figure 20: PCB Manufacturing Machine.	36
Figure 21: Drilling.....	37
Figure 22: Soldering.....	38
Figure 23: Printed circuits to manufacture the PCB.	39
Figure 24: System test in the breadboard.	41
Figure 25: Smart Lock System prototype working.	42
Figure 26: Universal Board Oscillator.	42
Figure 27: AVR Universal Board system test.....	43
Figure 28: Result of the hardware part.....	44

1 INTRODUCTION

The main idea of the Smart Lock System is to unlock a door with something used in everyday life as well as always is up to date, therefore; a Smartphone seemed most suitable. Everybody wants to have life as easy as possible, consequently carrying ten different hardware keys to open ten different doors are clearly out dated. This project changes the hardware key, as we know it and uses an Android app to substitute. It is not only an efficiency issue, but also rather a security matter. There are various advantages compared to the hardware key. For instance, the security factor as potential key losses are avoided since only one device, the smartphone, is required instead of multiple keys for the same task, as well as higher provision of security for one room or a whole building because only authorized personnel is able to access the area where the Door Lock System is installed.

The Industrial Innovation Academy (I2A) first proposed the topic. The main idea was to install the System in one of the rooms at the Technobothnia Lab, so only allowed students and teachers would have access. To achieve this, an Android App asks for a username and a password, if it is included in a database, the owner will be able to open the door using the app. Furthermore, this project was supposed to follow the thesis of another student; but finally it was impossible to have all his work and then the choice was to start over. It was possible to get some information about his previous work but not so much. Due to the fact to start over allowed more flexibility in the project and to create a new design. Moreover, it is important to say that this report was made to be as clear and helpful as possible, in case any future student wants to continue this work e.g.: improving it, will be able to understand the basis.

Therefore, the goal of this thesis is to create an electronic device, which will be able to control the status of the door strike, locking or unlocking it when appropriate by using an application in the Smartphone. The communication will be via Bluetooth. This channel of communication was chosen because nowadays most electronic devices, which use Bluetooth, can work with microcontrollers and the majority of smartphones supports Bluetooth.

The Smart Lock System is split into two parts: One is the software part, for developing the Android app to use as a key and the source code for the microcontroller. The other is the hardware part, which this thesis will be focused on. The design of the embedded system will be carried out with PADS, a program useful to perform PCB's because it provides simple tools and an easy usage, for doing the schematic and layout. In order to develop the lock system material provided from Technobothnia lab will be used such as the board, the vacuum UV exposing device, the PCB manufacturing machine and etc.

This thesis report is divided in different chapters. After the introduction, follows Chapter two, which outlines the smart lock system overview. There, the structure of the project will be explained, from the hardware point of view. The main chapters of this report are Chapter three and four. In Chapter three, the embedded system design is explained. This chapter will talk about choosing all the components for performing the embedded system, and about making the whole circuit. Chapter four focuses on the implementation of the embedded system, where PADS program is introduced and all the manufacturing PCB process is explained. Chapter five includes the system tests that have been done and all the results. At last, Chapter six presents the conclusions of the Smart Lock project, and possible improvements that could be added in the future.

The Smart Lock System could be a very useful choice for a company, which has different working areas and want to implement a security device in order to control the personnel access to certain areas. All the workers would have their own username and password. The Smart Lock System will be programmed to allow or deny access to certain areas for certain workers. Furthermore, it is possible to add a monitoring system to keep track of all the users' activity. Moreover, the client has to be informed that the Smart Lock System is an efficient, reliable and secure device, which can be implemented and used everywhere, without having to change the previous structure of the door too much.

2 SMART LOCK SYSTEM OVERVIEW

This chapter describes the global system architecture and the structure of the hardware part of the door lock project.

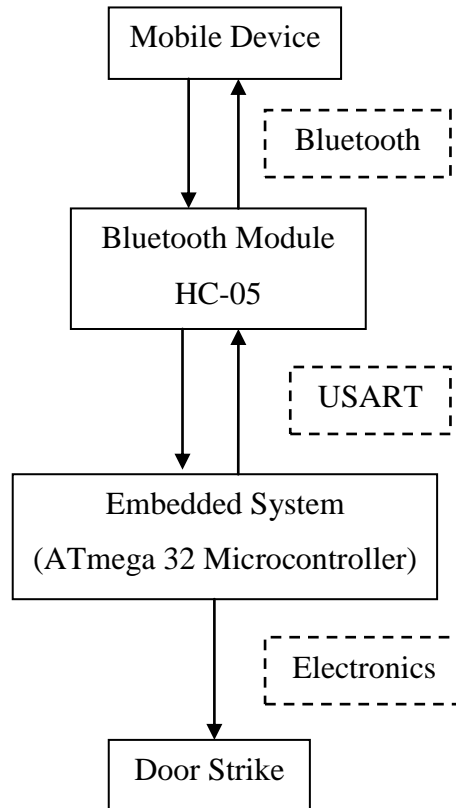


Figure 1: System overall structure.

As shown in figure 1, the system has different parts. First of all, the mobile device; it can be any device that supports Bluetooth such as smartphones, tablets, etc. In this project, an Android Smartphone will be used, as Android App will be developed to send and receive data to the Bluetooth module.

The mobile device will communicate via Bluetooth with the HC-05 Bluetooth Module. It is used to connect to another device in order to pass data wirelessly. Moreover, this device will be electronically connected to the embedded system, particularly to the microcontroller, which will be the Atmega32. Then, the communication between the mobile device and the embedded system that will allow the user to send and receive data both ways is possible. That is the reason why

there are two arrows between the mobile device and the Bluetooth module, as well as between the Bluetooth module and the microcontroller, because it will be possible to send the data to the microcontroller and to receive feedback from the microcontroller telling the user (to the mobile device) if the door has been opened or the permission has been denied.

In order to communicate with the Atmega32 microcontroller, the HC-05 Bluetooth Module sends the data via USART protocol. At this stage, the microcontroller, which has received the data from the HC-05 Module, will decide whether to allow or to deny the access to the owner of the mobile device, depending if the data received is valid or not.

The last step is to make the door strike to be locked or unlocked. If the received data from the mobile device through the HC-05 Bluetooth Module is correct, the microcontroller will send the signal to the door strike unlocking it just for a few seconds, allowing the user to open the door. As shown in figure 1, the communication between the microcontroller and the door strike will be through an electronic circuit.

2.1 Hardware Structure

This chapter is going to briefly explain the hardware part of the Smart Lock System. Greater details will be explained in the following chapters. The next image shows the whole electronic system.

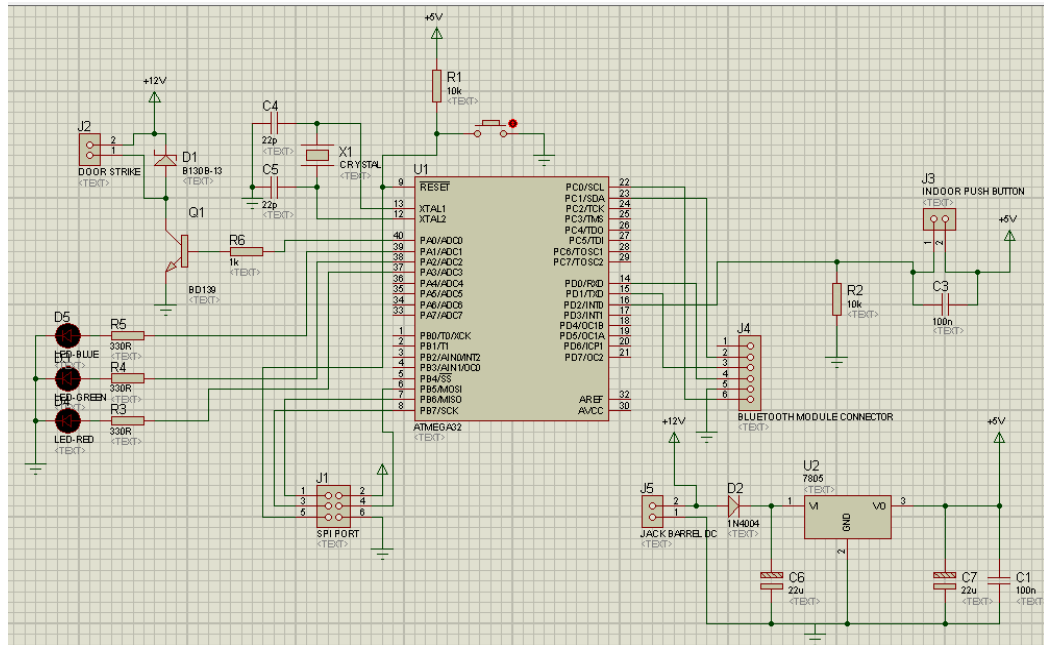


Figure 2: Schematic of the embedded system for the Smart Lock.

The software used to make this electronic circuit is Proteus Professional, which is based in two main devices, ISIS and ARES. The first one will perform the generation of the real circuits, and the second one will take care of the components in order to do the PCB. This program is useful due to the possibility to generate simulations. However in this case, it was impossible to generate simulations, as the circuit needs a Bluetooth module and an external device in order to make the microcontroller receive the messages from the Smartphone through the Bluetooth Module.

The picture shows all the parts of the circuit:

First of all and most importantly the Atmega32 microcontroller, which will handle all the communications, such as to receive and to send signals to all the parts of the circuit, in order to make it work.

The small circuit placed in the lower right corner is the power supply circuit. The input of the whole system will be 12V. The output of this power supply circuit is always 5V, so that the door strike will operate with the 12V and all the rest components with 5V.

There will be a 6-pin single row socket to connect the HC-05 Bluetooth Module and a 6-pin dual row socket for connecting the AVRISP mkII, which will be used to connect the PC with the microcontroller. This 6-pin dual row socket is usually called SPI port.

On the right side in the figure, there is a socket where the indoor button will be connected to allow the user to open the door at any time to leave the room. There is a capacitor parallel to it. This is placed in order to debouncing, which will be explained later.

On the top of the image, there is another switch button. This one can be used to reset the microcontroller, and thereby, the entire system.

The door strike will be connected to the 2-pin socket on the left side in the figure. This one is connected to the microcontroller using an electronic switch with an NPN transistor.

In order to know the state of the system and the door strike three LEDs were added. The first one to know if the system is working; the second one, a green LED, to know if the password has been accepted and the door opens; and the third one, a red LED, to know if the access has been denied.

Finally, the circuit also includes an 8MHz external oscillator for the microcontroller, formed by a crystal and two capacitors.

Everything said so far, corresponds to the embedded system design. After the design, the system has to be implemented. PADS program will be used in order to find a layout for the system, which will be printed on transparent paper. The next steps to obtain the real PCB will be done with the materials and machines provided by the Technobothnia lab. These steps will be explained in later chapters.

3 EMBEDDED SYSTEM DESIGN

The next picture shows the circuit, which will be implemented to create the Smart Lock System.

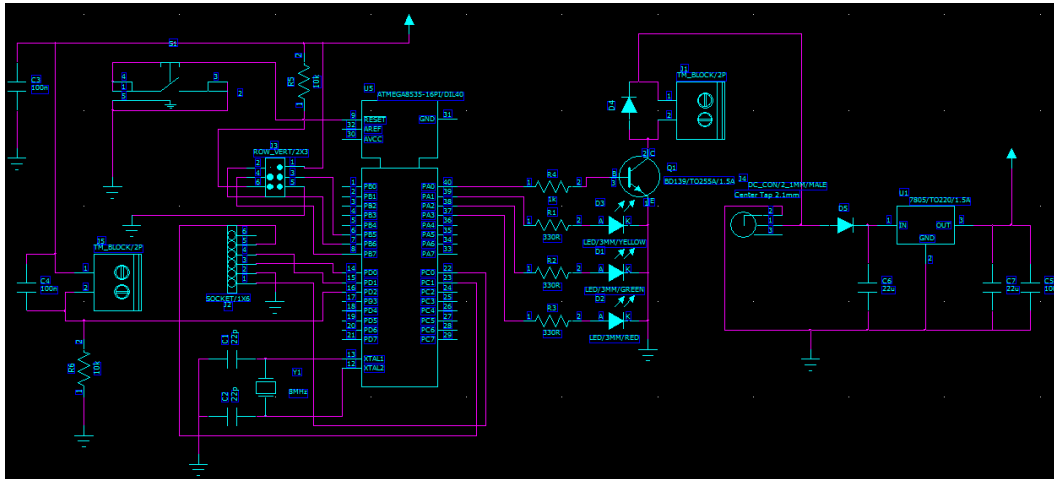


Figure 3: Schematic of the electronics of the door lock system.

This chapter describes the different parts to form the whole system.

3.1 Atmega32 Microcontroller

A microcontroller is a “System on a Chip” (SoC), which is a mini computer on a single integrated circuit. The microcontroller of this project is the “brain” of the system, as it controls the unit. All the tasks the system has to do always go through the microcontroller. It handles all the communication between all the parts of the system. Further, it is used to read the data sent from the Bluetooth Module, check if the password received is correct or not, send a feedback message to the Bluetooth Module, unlock the door strike when necessary, or switch on one of the three different LEDs. Hence, it can configure the Bluetooth module using AT commands, which are sent via USART communication. In order to switch the Bluetooth module to AT mode, the microcontroller activates (giving 5V) an output pin, which will be connected to the Bluetooth module key pin that will make the Bluetooth module convert in AT mode. Finally, the microcontroller influences the SPI port to make changes to the program or with the reset button reset the microcontroller.

The Atmel Atmega32 is the microcontroller used in this project. All its features are in the datasheet provided by Atmel (see References), however the remarkable characteristics for this project are:

- 4.5V – 5.5V Operating voltages
- SPI Serial Interface
- Speed Grade: 0 – 16 MHz

Some word clarifications that might appear in the datasheet: AVR is architected of an 8-bit microcontroller developed by Atmel. PDIP, also known as DIP or DIL, refers to an electronic component that has a rectangular housing and two parallel rows of electrical pins. MIPS means million instructions per second. USART, Universal Synchronous/Asynchronous Receiver/Transmitter, is the name of the communication language between the microcontroller and the Bluetooth module.

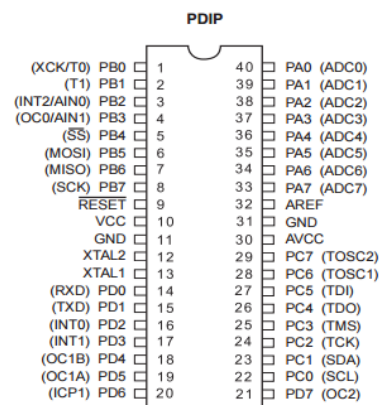


Figure 4: Pinout Atmega32.

The previous image shows the Atmega32 pin configuration. Although all the pin descriptions are in the datasheet, this report includes a brief summary of the pin descriptions used in the project:

All the Ports, A, B, C and D are 8-bit bi-directional I/O ports with internal pull-up resistors. Port B (pins from 1 to 8) also has some special features functions in the Atmega32. In this project, pins 6 (MOSI), 7 (MISO) and 8 (SCK) are needed for implementation of the SPI port, which will be explained in its own chapter. Pin 9 corresponds to the reset input; its functionality will be explained later. Pin 10 cor-

responds to VCC. This means that pin 10 always connects to the VCC (digital supply voltage), in this case 5V and pin 11 corresponds to the GND, the ground. The same as pin 10, this pin is always connected to the ground. Pin 12 is the XTAL2, which is the output from the inverting oscillator amplifier. Pin 13 corresponds to XTAL1, which is the input to the inverting oscillator amplifier, and also the input to the internal clock. These two pins will be explained deeply in the oscillator chapter.

In the datasheet (see References) is much helpful information about the Atmega32.

3.2 HC-05 Bluetooth Module

Bluetooth is a wireless technology to exchange data over short distances between fixed and mobile devices. Nowadays, there are a lot of Bluetooth modules and plenty of them could be suitable for make this project work. In the Technobothnia Lab, they had the HC – 05 Bluetooth Module, consequently this device will be the one this project uses for the implementation of the Smart Lock System. In its datasheet (see References), all the features of this device are explained. The advantages of this module are simplicity and easy usage. For this reason, this module is commonly used in AVR microcontroller's and Arduino's projects. Also, it is important to say that the HC-05 is a Master-Slave module, which means that in addition to receiving connections from a PC or a tablet, it is able to generate connections to other Bluetooth devices, too.

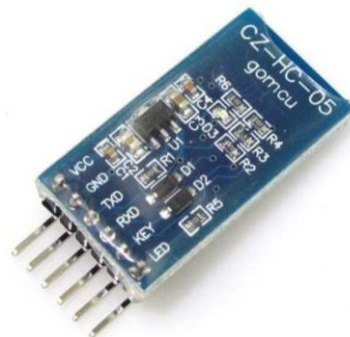


Figure 5: HC – 05 Bluetooth Module.

This Bluetooth SPP (Serial Port Protocol) module has as well remarkable hardware features:

- Low Power 1.8V Operation, 1.8 to 3.6V I/O
- With edge connector (row of contacts to facilitate connection to external circuits)

As the previous image shows, the Bluetooth Module has 6 pins. Pin VCC, as its name indicates, has to be connected to the power. In this project, it will be connected to an output pin from the microcontroller, pin PC0, which will give power to the module. Although, the module runs on 3.3V power, it also supports operation at 5V power, which is the output voltage of the Atmega32 (VCC). Pin GND will be connected to the ground of the embedded system. Through pin TXD, HC-05 module will be able to transmit serial data from itself to the microcontroller. This pin has to be connected to the RXD pin (PD0) of the Atmega32 in order to establish communication between them: one device will transmit data, and the other one will read the data. The same model applies for the RXD (Receive Serial Data) pin; it has to be connected to the microcontroller TXD pin (PD1) in order to be read by the Bluetooth module transmitted by the microcontroller. KEY pin is used to start the Bluetooth module as AT mode or normal (master-slave) mode. AT mode is to configure the module, changing internal settings. Then, the user will send some formatted commands to set the Bluetooth module parameters to his or her needs. In order to start running the module with the AT mode, the KEY pin must be powered on before the VCC pin of the module gets power. That is why the VCC pin of the module is connected to a microcontroller output pin and not directly to the power of the whole system, otherwise it is possible to add a software delay in this pin to achieve that the KEY pin gets powered before the VCC pin, in the case the user wants to start the module in AT mode. Further, the hardware part just has to connect the KEY pin to a microcontroller pin, so that the software part of the project will handle the other. Last pin of the module is the LED pin (or STATE pin), which tells if the device is connected or not. In this project, this LED pin is not needed.

With this, it is possible to establish communication between the microcontroller and the Bluetooth module. The module will receive the data from the users' Smartphone and through the TXD pin, the data will be sent to the RXD pin of the microcontroller, allowing it to do the necessary data checking and processing. Then, the microcontroller sends the data through the TXD pin to the RXD module pin to enable the module to send via Bluetooth, feedback to the users' Smartphone. It is also possible to start the Bluetooth module as AT mode to change the internal settings of the module.

3.3 Door Strike

This subchapter describes the whole circuit of the door strike in order to work. Basically, this part is based on the resistor, a transistor, which will work as a switch, a Schottky diode to protect the circuit, and a socket where the door strike will be connected.

The point of using a transistor, as a switch is that an output pin of the microcontroller, with a voltage of 5V, controls its base and the circuit that makes the door strike work is powered with 12V. Then, the advantage is to control the 12V circuit for the door strike without using too much energy; said in another way, the small base current of the transistor controls a much larger collector load current. It is only required to overcome the $0.7 V_{BE}$ of the transistor in order to cause base current flow, which consequently will control the door strike allowing it to be unlocked. So, it is possible to say that the transistor is used as a switch and as an amplifier: it uses a relatively low power signal in order to switch a circuit, which works with a relatively large amount of power.

The NPN transistor is chosen depending on the load. The most important specifications to look for are the maximum collector current, I_C , and the current gain, h_{FE} . It is not the same to use a BC107 that allows currents up to 100mA, than a 2N3055 that allows currents up to 15A. Therefore, it is very important to know the current, where the transistor will be placed in.

In this project, the door strike used is the ALCAD ABR-001, a simple electronic



Figure 6: Electric door strike used in this project

lock. Simple or standard electronic locks are the ones which enable opening the entrance door when they receive the 12V voltage impulse. It remains released as long as the button is pressed on the subscriber's monitor, in this project, as long as the microcontroller gives power, which is 4 seconds. There is another type of electric door strikes, the automatic

ones. They stay active until the door is closed, therefore it is not necessary to continue pressing the button, and in this case it would not be necessary to open the door within the 4 seconds. One example of this type of electric lock is the ABR-003, from the same manufacturer as the ALCAD. Due to the goal to provide security and safe power, this kind of electric lock was discarded. The ALCAD ABR-001 is a fail-secure strike, this means that it is unlocked if applying electric current, and in the case of power failure this device would be remained locked.

From the webpage of the door strike manufacturer, it is known that this device needs a power of 12V to work. And in the lab, the parameters while it was connected were measured and the results were as follows:

- Impedance of the door strike: 11 Ω
- Current through it: around 0.5 A

Coming back to the choosing of the transistor, it requires one that I_C max is more than 0.5 A. There are plenty of transistors, which satisfy this specification, so there are a lot of different possibilities. In this project, transistor BD139 was used. Some of its absolute maximum ratings and specifications according to its datasheet (see References) are:

- Maximum Collector-Emitter Voltage, V_{CEO} : 80V
- Maximum Collector Current (DC), I_C : 1.5A
- Base-Emitter On Voltage, $V_{BE(on)}$: 1V
- And its DC Current Gain, h_{FE} (measured with tester): 110

The transistor used by this project is, specifically, the BD139G, where G means Pb-free package, and this refers to the semiconductor electronic products that are compatible with the current RoHS regulations, which are created to eliminate or limit substances that create pollution or are dangerous to the environment and to the people. One of these regulations establishes that there must not be more than 0.1% of lead in homogeneous material (by weight).

Once the transistor was selected, the resistor between the microcontroller and the base-transistor pin has to be calculated. The value of the base-transistor resistor depends on the project, i.e. it depends on the currents and voltages that the load will need. Normally the base resistor is much bigger than the collector load (because the current in the base is very small). It is very important to calculate, simulate and measure the parameters in order to achieve correct outcomes. According to the next formula, which is the modified version of the Ohm's law, it is possible to calculate the value of the required resistor:

$$R = \frac{V - 0.7}{\frac{I}{h_{FE}}}$$

Equation 1

Where V is the voltage that the microcontroller gives and 0.7 is the voltage drop between the base and the emitter of the transistor, then, the numerator of the equation, " $V - 0.7$ ", is the voltage drop through the resistor. Continuing with Equation 1, I is the current that flows through the door strike, and h_{FE} is the current gain of the transistor. Then, on the denominator of Equation 1 there is the base current of the transistor calculated according to the transistor general equation:

$$I_C = h_{FE} \cdot I_B \rightarrow I_B = \frac{I_C}{h_{FE}}$$

Equation 2

Then, according to the Ohm's law, $R = \frac{V}{I}$ (where V is the voltage drop between both terminals of the resistor), the resistor value can be calculated using the equation 1:

$$R = \frac{5 - 0.7}{\frac{0.5}{110}} = 946 \Omega$$

In order to be stricter, the resistor chosen has a value of $1K\Omega$, resistor which will be placed between the PA0 pin in the microcontroller and the base pin of the transistor in order to get a very small value of current through this part of the circuit. If the output of the microcontroller is 5V, V_{BE} of the transistor is 0,7V and the value of the resistor is 1000Ω , the current through the base can be calculated according to the Ohm's law, $I = \frac{V}{R}$ (where V is the voltage drop between both terminals of the resistor). This current should be:

$$I_b = \frac{5 - 0.7}{1000} = 4,3 \text{ mA}$$

In the figure of the whole system is a Schottky diode parallel to the door strike. It is highly important to place the diode there, but first it is necessary to know how the internal mechanism of the electric strike works. This internal mechanism, which controls the operation of the door strike, is different according to the manufacturer. Most of them are controlled by a solenoid which operates a mechanism that will control the keeper or jaw. A solenoid is basically a coil formed by a spirally wound wire on a cylindrical frame, which creates a magnetic field when a DC current flows through it. In this project, due to the fact that the current flows through the coil, a self induced magnetic field is pulling the electric lock mechanism. When the DC power that supplies the coil is turned off (the current in the coil is turned "off"), a huge back electromotive force (emf) voltage is created as the magnetic flux collapses inside the coil. This induced reverse voltage can be very high, normally around 10 times higher than the nominal voltage of the electric lock; this project uses a 12V DC lock, so it could reach 120V DC. This reverse voltage is powerful enough to destroy the controlling device of the door strike, which is the BD139 transistor, because looking at the specifications of this device, the $V_{CEO \text{ max}}$ is 80V, so 120V would destroy the transistor. Now, the idea of the diode appears. A diode in parallel to the lock coil will short out the high reverse voltage in order to stop this high voltage to go through the transistor de-

stroying it. Then, the coil will dissipate its energy through the diode. To sum up, the diode is used to protect the BD139 transistor from the high reverse voltage that is developed by an electric lock's coil when the DC power is turned off. The diode must be connected in reverse polarity, its cathode to the + and its anode to the -. To chose the correct diode, it is important to know the maximum current that the electric strike consumes, in this case 0,5 A to ensure perfect handling. Moreover, its reverse block voltage must be twice the peak of the possible reverse voltage, in this case 120V. Therefore, the 1N4004 diode would be the perfect choice because it can handle, according to its datasheet (see References), currents up to 1A and its reverse block voltage is 400V. This project only considered by mistake the first tip that the diode must be able to handle the maximum current of the electric strike, and therefore the choice was to use the SB130 Schottky diode. From its datasheet (see References), the relevant specifications for this project are as follows:

- Maximum DC reverse blocking voltage, V_{DC} : 30V
- Maximum average rectified output (forward) current, I_O , $I_{F(AV)}$: 1 A

The underlying idea to choose the Schottky diode was because it has a very fast switching action between direct and reverse states, which allows removing the excess of current. However, as said before, the 1N4004 diode could work in a more suitable way.

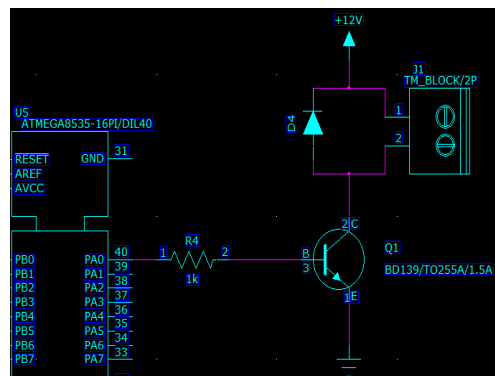


Figure 7: Door strike circuit.

Now that the main two things of this circuit (using a transistor as a switch and using a diode in parallel with the door strike to give protection to the circuit) have

been explained, it is possible to summarize what this part does in general. When the user is allowed to enter the room, the microcontroller gives power ($VCC=5V$) to PA0 pin connected to a resistor to make the current as low as possible before going to the base pin of the transistor, the next component. When the power arrives at the base pin of the transistor, this device allows current flow between its collector and emitter pins, closing thus the electric strike circuit. Once this circuit is closed, current can flow from the power supply of 12V through the door strike and transistor to the ground, thereby the electric strike internal mechanism allows the keeper to be unlocked during the time that the microcontroller provides power (4 seconds). This is the time when the user can open the door. When the PA0 pin no longer provides power the transistor is cut off, and then the 12V DC power is turned off too and the coil inside the electric strike must dissipate its energy through the diode to avoid the destruction of the transistor. It was impossible to find the datasheet of the door strike used in the project, so it is not possible to explain exactly how the internal mechanism of it works.

3.4 Power Supply Circuit

This subchapter is focused on the power supply circuit of the whole system. The project will implement a system, which could power all the components with 5V and the electric strike with 12V.

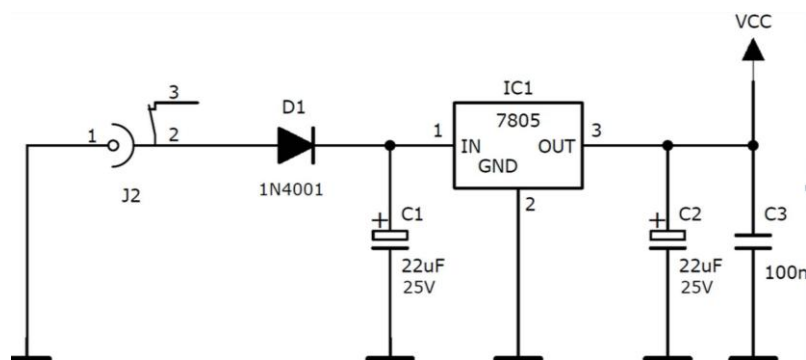


Figure 8: Power supply circuit.

The key to implementing this circuit is using an IC 7805, which is a voltage regulator integrated circuit. It can accept an input voltage between 5V to 18V and it always provides 5V regulated power supply. In its datasheet (see References)

there are a lot of features and specifications, but the most relevant for this project is the output voltage of 5V. Its datasheet also includes a chapter of typical applica-



Figure 9: DC barrel power jack

tions, where the circuit that this project is implementing appears. The first component in that circuit is a DC barrel power jack/connector, where the connector that provides 12V will be plugged in (An AC adapter plugged in the electricity network that provides 12V of DC current). It has three pins but in

this project just two of them are used, the + and the – of the power supply. After this component, the voltage is 12V DC, so, from here there will be a constant connection to the door strike. Next component is a diode. It could be avoided but it is placed just for security. It should be a 1N4001 (see References for its datasheet) that can handle currents up to 1A and its reverse block voltage is 50V, but in the Technobothnia Lab only 1N4004 were provided (see References for its datasheet), so this last one was used. The only difference is the higher DC blocking voltage of the 1N4004, in particular 400V. In the diode is a voltage drop of 0.7V, and then in the input of the IC7805 there are 11.3V. Before and after the IC7805 there are three capacitors. The one placed in the input of the IC7805 works as a filter, employed to steady the slow changes in the input voltage of the circuit. A good value of this capacitor is 22 μ F in order to have an acceptable stabilization. The second capacitor, the one in the output of the IC7805 also works as a filter, used to steady the slow alterations in the output voltage. A large value of the capacitor amplifies the stabilization, but a suitable value is also 22 μ F. The last capacitor is known as a bypass capacitor and it is used to bypass very small period spikes to the earth without influencing the other components. And finally, the IC7805 which has three pins. In the input pin, the unregulated voltage (11.3V) is given. The ground pin is equally neutral for the input and the output. And the output pin provides the regulated voltage, 5V, where all the components in exception of the electric strike will take connection from here.

Thus, the advantage of this circuit is that the system does not have to implement two separate circuits, one for the power supply of the components and the other

one to the power supply of the electric strike. In addition, providing the system with two different batteries, one of 5V and another one of 12V, is avoided.

3.5 Indoor Push Button

This subchapter explains the circuit implemented in order to place an indoor push button in the circuit. Two important parts to implement this circuit will be explained: the capacitor parallel to the push button and the resistor.

All doors can be open from the inside without using a key, because the door lock security is only acting entering the room or building. This project needs an electronic circuit that implements the indoor push button in order to unlock the electric strike from inside without having to use the Android App. This simple circuit is shown in the next figure:

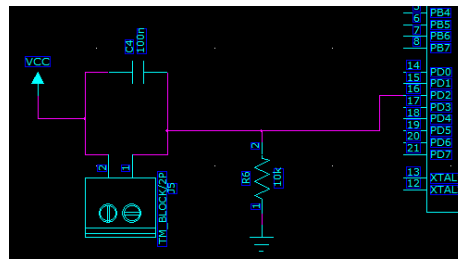


Figure 10: Circuit for the indoor push button.

A socket is used instead of directly placing the button, because when the system will be implemented in a real door, perhaps it will be better to place the system somewhere far from the door. Then it will be possible to connect the button in the socket and place it where appropriate, but next to the door. Also, the goal of the socket is to be able to connect a good and big push button, not the small ones that are in the Lab.

In the figure is a capacitor parallel to the push button in order to debounce the switch. Bouncing is the tendency of two metal contacts generating multiple signals as the contacts close or open. For example, when a key on the computer keyboard is pressed, people expect just one contact to be recorded by the computer, however, it exists the initial contact, after there is a slight bounce of the contact, then another contact when the bounce ends, later another bounce back, etc.

Debouncing is the process of removing bounces. It can be done by some kind of hardware device or software (in this case by hardware, a capacitor), that makes sure that just a single signal will be registered for a single opening or closing of a contact. This project uses a capacitor in order to do the debouncing. This way is the simplest way to remove the bouncing instead of using a capacitor and a resistor for getting the perfect timing. It will create a nice sloping effect removing the bouncing. This project uses a 100nF capacitor, appropriate for its purpose.

The resistor used has also an important role in the circuit; it is a pull-down resistor. This resistor holds the logic signal at the low logic level (0V) when no other active device is connected, in this case, when the button is not pushed the voltage in the resistor will be 0V. If this resistor was not there: when the button is not pushed and the program tries to read the state of that input pin; will it be high (pulled to VCC) or low (pulled to GND)? It is not known. This event is called floating. So the resistor is placed there to prevent this unknown state.

If the user pushes the indoor button, then the current flows through it until arriving to the input of the microcontroller, in this case pin PD2, which will be programmed by the software part to provide power to the pin PA0, which will unlock the electric strike.

3.6 SPI Port

“Serial Peripheral Interface (SPI) is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances. With an SPI connection there is always one master device (usually a microcontroller) that controls the peripheral devices. Typically there are three lines common to all the devices:

MISO (Master In Slave Out) – The Slave line for sending data to the master.

MOSI (Master Out Slave In) – The Master line for sending data to the peripherals.

SCK (Serial Clock) – The clock pulses which synchronize data transmission generated by the master.” (Arduino official webpage – Reference – SPI)

But why it is needed? Why not just use a serial port with a TX and a RX? A common serial port with a TX and RX lines is an asynchronous port because there is no guarantee that both sides of the communication are running at the same rate, due to differences in their clocks, consequently both sides must agree in advance on the transmission speed. However, a SPI port is a synchronous data bus, which means both sides are in perfect sync due to the fact that SPI has the lines for data and clock separately, then, the clock can tell the receiver the accurate moment to sample the bits on the data line. The side, which generates the clock signal is called Master (SCK pin is the clock output), and the other side is the Slave (SCK is the clock input pin). To send and receive data through both sides, lines MISO and MOSI (named before) are used.



Figure 11: Connecting SPI Port.

The previous image shows the way that a SPI port has to be connected to the microcontroller. As said in the software structure, this port will allow the microcontroller to be programmed. In order to achieve this, it uses a device called AVRISP mkII, shown below, which is connected via USB to the PC and via the SPI port to interact with the microcontroller. After that, it is able to upgrade the Atmel 8-bit AVR microcontroller by passing data through it.



Figure 12: AVRISP mkII.

3.7 Other parts

Here, the rest of the parts in the circuit are explained.

3.7.1 Oscillator

The internal oscillator of the Atmega32 has a maximum speed of 16MHz. In this project, it is thought to be running at 8MHz, so there is no need to implement an external oscillator. However, the internal RC oscillator of the microcontroller is more sensitive to voltage and temperature changes. Hence, an external crystal will be more stable than the internal oscillator, and more accurate.

In the Atmega32 datasheet (see References), there is an explanation of the crystal oscillator etc. According to it, XTAL1 and XTAL2 are, respectively, the input and the output of an inverting amplifier, which can be transformed for usage as an “On-chip Oscillator”, as the figure on the right shows. It can be used as a quartz crystal or a ceramic resonator. Atmega32 datasheet also provides a table to choose the correct value of the capacitors. Its value depends on different factors.

However, the values of C1 and C2 should always be equal. This project implements C1 and C2 of 22pF.

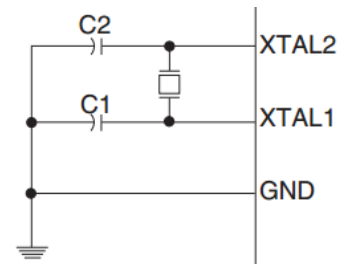


Figure 13: Crystal Oscillator Connections.

3.7.2 LED's

This project also implements some LED's (Light Emitting Diode) to let the user know the state of the door lock system.

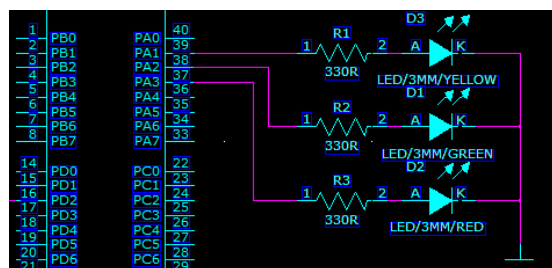


Figure 14: LED's used to show the system state.

The peak forward, the current of a normal LED is approximately 100mA. Therefore, placing a resistor between the LED and the microcontroller pin is required,

in order to avoid destruction of the LED. A normal resistor of 330Ω should protect this device. If the output voltage of the Atmega32 is 5V and the voltage drops in the LED is approximately 3V (forward voltage of a LED), then, according to the Ohm's law, $I = \frac{V}{R}$ (where V is the voltage drop between both terminals of the resistor):

$$I = \frac{5 - 3}{330} = 6 \text{ mA}$$

Now, it is impossible to destroy the LED.

When the system is plugged in, the microcontroller will output 5V through PA1 pin, illuminating the working LED, in this case, white. When either the password has been accepted or the indoor button has been pushed in order to open the door, Atmega32 will provide power in the PA2 pin and consequently the green LED will turn on showing that the door can be opened. And finally, if the user tries to open the door with the Smartphone and he or she is not allowed access, then the microcontroller will give 5V to its PA3 pin allowing the red LED to be illuminated.

3.7.3 Reset Button

All the circuits must have a reset button that can reset the device if crashed. To implement this button, it is important to know that the Atmega32 has the Reset input as \overline{RESET} , so, a low voltage level on this pin will generate a reset to the microcontroller. Thus, in a normal state, a high voltage level should power this pin. The design should be as the figure in the right side shows.

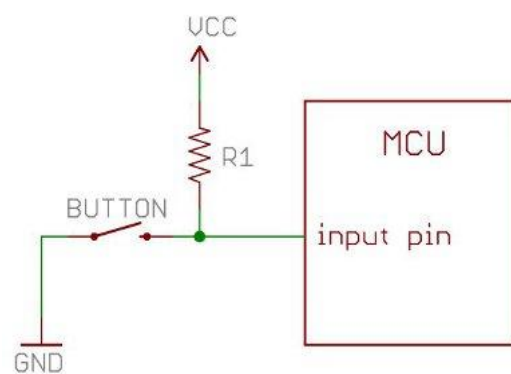


Figure 15: Reset button design

The philosophy of the resistor is the same as the pull-down resistor explained in chapter 3.5. This resistor will ensure that the input pin of the microcontroller is in a high low state, and at the same time, using a low amount of current. If the resistor were not placed there, VCC would connect to ground when the user pushes the button, which would cause a short. A suitable value for this resistor has to be on the order of $10k\Omega$.

In a normal mode, with no reset, pin 9 of the microcontroller is connected through a resistor to the VCC. As this input pin is “No Reset” and there is a high voltage level in it, the Reset is now set to a low voltage level, so there is no reset to the system. When the user push the button, pin 9 is connected directly to the ground, then the system does a reset. The datasheet of the microcontroller (see References) also explains that there is a minimum pulse length in order to generate the reset; and there is no need for the clock running to generate the reset.

3.7.4 Capacitor in pin 10

The idea of placing a capacitor in the pin 10 of the microcontroller (VCC) to the pin 11 (GND) is very simple. This device works as a stabilizer, filtering the two pins: the VCC and the ground. It gets rid much noise that may be introduced to the system. The standard value for this kind of capacitor is $100nF$.

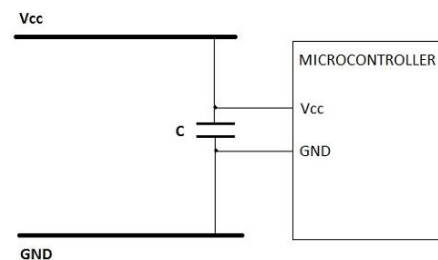


Figure 16: Stabilizer capacitor

4 EMBEDDED SYSTEM IMPLEMENTATION

Once the design of the system is finished, the implementation can be carried out. The first step to do the implementation is to choose a program in order to obtain the prototype. As mentioned before, the program that will be used in order to achieve the embedded system is PADS.

4.1 PADS PCB Design Software

In order to create the layout of the system circuit, this project will use PADS PCB Design Software, a program that provides intuitive and affordable methods to design the printed circuit board, providing the necessary tools for schematic, layout and rapid prototyping. In order to obtain the result, it is needed two sub software of PADS, first PADS Logic, and second PADS Layout.

4.1.1 PADS Logic

PADS Logic is the software that allows the user to create the schematic of the system. It provides everything needed for the circuit design, component selection, library management, etc.

After creating a new file, a black sheet will appear in the screen to place all the components and connections. In the schematic editing toolbar there are some options, but the most important ones are the Add Part and Add Connection buttons. In the Add Part window, select a library to choose the right component. There are many, however choosing one from the libraries from VAMK, the user can be sure these components will be in the Technobothnia Lab. First, the user has to add all the components with the Add Part button placing them in the sheet. For example, if a resistor is needed, it will be in the `vamk_lib\VAMK_R`, or an analog device such as a transistor, it will be placed in the `vamk_lib\VAMK_ANA`.

After adding all the components, next step is to make all the connections between them. The button to do this is the Add Connection button. Then, with the cursor, the user must click in the required pins of the components to make the connection, as shown in figure 5. It is very important that the user checks that he/she will not

make any mistake in this part, because a mistake here will make the circuit useless. If the user does not realize the error until he/she runs a test before the PCB is already manufactured, the user will have wasted a lot of time performing the PCB for nothing. Hence, it is better to be very careful doing the connections and checking all the time that there are no mistakes. Once placing all the components and connections, the result should look as Figure 17:

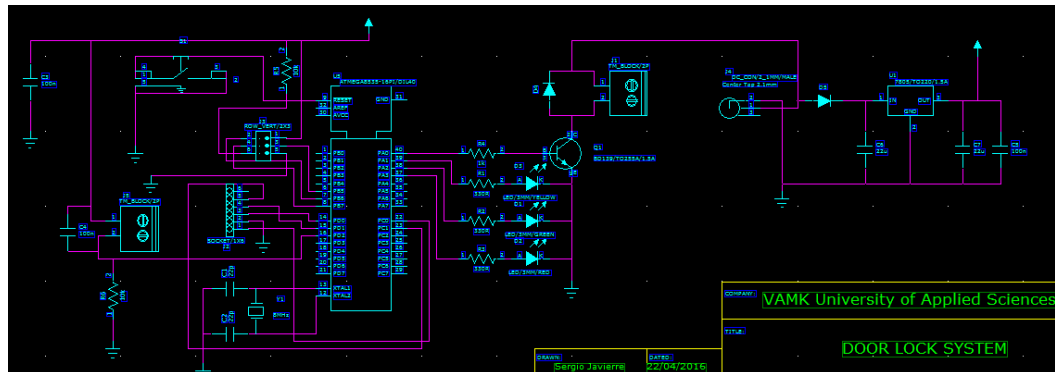


Figure 17: Result of PADS Logic design.

As the figure shows, the microcontroller used in PADS is the 8535, but this project will implement ATmega32. It is possible to use 8535 in PADS because the pin out of both microcontrollers is equal.

Next step is to create the Layout Netlist in the Tool menu that will create the PCB net list file, which will be read by the PADS Layout. At this point, it is also necessary to check if a file with the signature .err is not created. This will tell the user if there are any errors in the design.

After creating the net list, the user has to send it to PADS Layout, which can be done in Tool menu. Sometimes the server is busy and the user must click on “Switch To...” button until the PADS Layout can be connected. Afterwards, in the PADS Logic window another window will appear, called PADS Layout link. There, the user must click the “Send Net list” button in the Design menu, which will generate the net list and exports it to the PADS Layout. At this point the user can start working with the PADS Layout.

4.1.2 PADS Layout

With the help of the PADS Layout the user will be able to get the final prototype for the PCB of the system. Now, the user has to move all the components in a specific place to make all the connections without any of them crossing each other otherwise a short will occur. If it is not possible or the user does not find a way to do this, it is always possible to create a via, which will allow the user to do the connection in the other layer. This project has achieved to perform the connections of the PCB in just one layer and to make the PCB as small as possible in order to waste less copper.

All the components have information about themselves in the screen by default, which makes working on the project more difficult. However, there is an option, the Display Colors in Setup menu, which will allow the user to remove all the names and descriptions.

Once the user thinks that everything is in the correct place, the next step is to draw the connections. The button to do this is the Add Route in the Design toolbar. Note: by default, all the components are in the Top layer. Before starting to draw the connections, the user has to change the layer from Top to Bottom in order to have the Bottom layer as the copper layer as all the connections will be on there. If the user needs to place a via, that connection will be placed in the Top layer. Before drawing the routes, it is necessary to know that there should not be any routes with a 90-degree turn, because this would use longer traces (higher impedance → more copper usage → larger area) and the likelihood to fetch problems during manufacturing increases. Plus, before making the routes, the user can specify the trace width and the clearance settings by going to Setup → Design Rules → Default → Clearance. The most important settings are trace width (Recommended), and clearance between Copper and via such as trace and pad. Further, the user is always able to change the width of a certain trace going to the properties of that specific line. In this project, the recommended trace width is set to 0.6, and the clearance between copper and trace as well as via and pad is set to 0.8 or 1. Some width lines have been changed to 0.5, 0.4 or even 0.3 in order to be able

to pass between two pins of the microcontroller or between the pins of the SPI Port, and thereby, avoiding placing a via.

When all the connections have been made, the next step is to draw the board outline. This button is located in the drafting toolbar. After the board outline, the user must add the copper pour, button “Copper Pour” placed in the drafting toolbar. After this, the user has to change some parameters in the pop-up window “Add drafting”. In the Flood & Hatch options the Hatch grid was set to 0.1 and the Smoothing radius to 0.05. After the Copper Pour, the user has to create the “Flood”. Once this is done, the result should be like the following image:

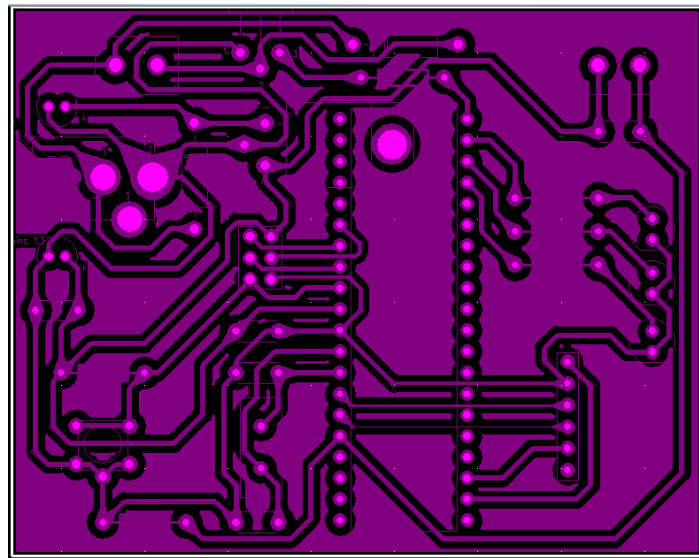


Figure 18: Result of PADS Layout.

At this point, the user can verify the design with the option “Verify Design” in the tools menu, which will tell him/her the clearance errors and others. Due to the fact that some components had some text and the body of the transistor was outside the board outline, there occurred some clearance errors in this project. These minor flaws were fortunately irrelevant for the outcome. Note that, if the user wants to modify the circuit after the flood, he/she can do it after using the Copper Cut Out button in the drafting toolbar.

Finally, the last step was to create a printable file. The user has to go to File and click on CAM option. In the CAM window, the user has to create a CAM Docu-

ment and edit it. The most remarkable settings are: Justification, centred in the Options Button, as well as in the Layers Button the user needs to add the layer(s) used to make the connections. In this case just the Bottom layer adding the items he/she wants to be printed, usually: Pads, Traces, Vias, Copper, Keepouts, Bottom Mounted, Board Outline and Connections. The last setting is to click on the “Print” button in the Output Device field. The user must be provided with a transparent paper and he/she has to put it in the printer paper tray, and then, in the CAM window, push “Run”. The user will get the circuit printed on the transparent paper, and with it, the user is ready to manufacture the PCB in the Technobothnia Lab.

4.2 PCB Manufacturing

This chapter describes all the procedures carried out to manufacture the real PCB. This process was performed in the Technobothnia Lab.

4.2.1 Provisioning

First of all, all the components are needed. In this case, they were taken from the cabinet located in front of the LEC-4 room in Technobothnia. There is a guide to allow the user find the components in their specific place. As said before, all the components found in the VAMK libraries in program PADS, should be available. After taking all the components, the next working place will be the room to manufacture PCB's. Lastly the user has to get the board where the connections and the components will be placed. These items are located inside the cabinet in the PCB manufacturing room. Note that there are one-side boards and two-side boards, which mean boards that will have one or two sides of copper. In this case, a one-side board is needed.

Once all the components have been picked up, and with the circuit printed in the transparent paper and the board, everything is ready to start making the manufacturing of the PCB.

The next table includes all the components needed for the implementation:

NAME	QUANTITY
ATmega32 Microcontroller	1
Capacitor 22pF	2
Crystal 8MHz	1
SPI Port	1
Push Button (Interruptor)	1
Socket 2 pins	2
Socket ATmega32	1
Resistor 10k	2
Diode 1N4001	1
Red LED	1
Blue LED	1
Green LED	1
Resistor 330ohm	3
BD139 Transistor	1
HC-05 Bluetooth Module	1
Switch Button	1
Socket Bluetooth Module	1
Resistor 1k	1
Capacitor 100nF	3
Capacitor 22uF	2
Diode 1N4004 (or 7)	1
IC7805	1
DC Jack Barrel	1
Door Strike ALCAD ABR-001	1

Table 1: Material List.

4.2.2 Vacuum UV Exposing Device

In order to achieve the goal of manufacture the PCB, this step is absolutely necessary. This gadget is equipped with special UV lamps and auxiliary vacuum suction device. The only things the user has to do is to remove the adhesive from the board, putting the transparent paper face up in the UV device and the board with the side where there was the adhesive face down, and then close both covers of the UV device. There are two covers, the inner one, a transparency film, to make the vacuum effect and the outer one which will make the UV exposing effect to the top layer if necessary, in this case not because the prototype implemented has only one side and it will be printed in the board using the bottom UV light. It is

important to place the board quickly in the vacuum UV exposing device because the board is very light sensitive and it is better not to be exposed to it too much.



Figure 19: Vacuum UV Exposing Device.

And also, the user has to be careful about placing the transparent paper and the board in the correct side; in case of mistake, it will be impossible to place the components later because the circuit will be printed in the wrong side.

The circuit has to be printed in the bottom layer of the copper plate. This means that the preview offered by PADS Layout will be printed inverted in the bottom part of the board, allowing the components to be placed in the top layer and soldered in the bottom layer, thus the circuit will be implemented in the proper way.

The circuit has to be printed in the bottom layer of the copper

After placing the board and the transparent film inside the machine, the user has to set the exposure time to 3 min and let it work. The board is formed by an insulating layer, then the copper layer, and finally covered with a light sensitive emulsion. Once the board is exposed to the UV radiation, the zones in the board covered by the black part in the transparent paper will have protected the emulsion from the UV radiation, and the emulsion in the other parts of the board which got UV, will be weakened. Finally, say that the vacuum is to keep the board next to the transparent paper, without any space between them.

4.2.3 PCB Manufacturing Machine

This machine completes the manufacturing process of the PCB. It has 5 steps, which are 5 holes with specific liquids inside where the board has to be submerged. First of all, the machine has to be turned on setting the time to the max (with the counter in the left of the machine), and then it will heat the liquid in the step 3. There are two lights in the machine, one means Etching and the other Heating. Etching light is always turned on while the machine is on. When the liquid has reached the set temperature (50 degrees, set in the counter in the right of the

machine), the Heating light will turn off and the etching process will be much faster.

The step zero is to sustain the board with a frame, and then the user can start the process. First step is to submerge the board in a water-based off-set plate developer solution, where the board will get rid of the weakened part of the emulsion. Then, the copper layer in these parts will be the outer layer, but the part of the board that was covered by the black part of the circuit in the transparent paper, will



Figure 20: PCB Manufacturing Machine.

be still protected by the emulsion. It doesn't need a lot of time, just a couple of minutes. The second hole is to wash the board with water. The third step is very important. This hole contains iron (III) chloride (ferric chloride), which will remove the most external copper layer, which corresponds to the transparent part of the printed circuit in the transparent film, leaving this part just with the insulating layer. The user has to keep watching the PCB while this step, and when the board color will be yellow instead of pink means that the PCB is ready. In the fourth and fifth holes there are water and water with some chemical component which makes the PCB clean and the copper ready to be used as wires. Also, a tin process is done in these last two steps to protect the traces.

The user has to be careful with these products because they are very harmful for the skin and for the eyes. Next to the machine, there is a medical kit in case of an incident to eye-flushing.

After this process, the PCB needs to be washed with tap water and, at this point, it can be removed from the frame, and finally it must be dried.

4.2.4 Drilling machine

After using the PCB manufacturing machine, the user can cut the PCB by the board outline. At this stage, the user can proceed making the holes. It is very im-

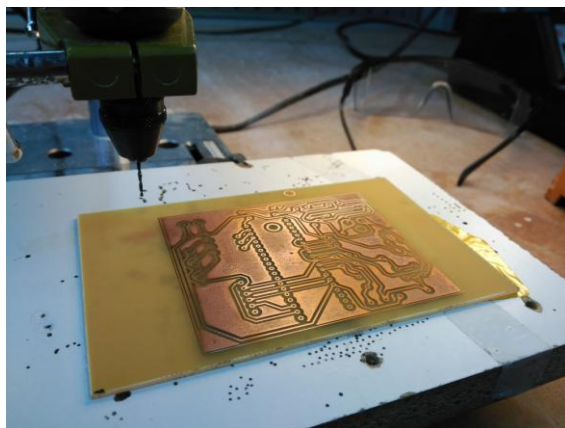


Figure 21: Drilling.

portant to use the protection glasses because during this process waste material can jump to the eyes. The PCB has to be placed over a wooden plate, and then the user can start drilling. Maybe some components have their pins bigger, and then the user has to change the drill to another one with large diameter so that the component can fit in its place.

After drilling, the copper layer of the PCB may be dirty. It is used alcohol in order to clean this layer. Also it is possible to polish the board borders with a rasp.

4.2.5 Soldering-iron

Now is the moment to place all the components and to soldering them in the board. Before doing this, the user can use a flux pen to mark the holes because thus, it will be easy for the tin to get stuck in the board. Finally, by looking how the component is placed, it has to be located in the correct holes and then it must be soldered there. Some tips to solder: use a soldering-iron with a sharp end, because there are pins which are very close to each other and a soldering-iron with a sharp end will help to don't do mistakes. To solder, first step is to heat the pad and the terminal of the component with the soldering-iron during some seconds, and then add tin to the pad until the pad is completely covered of tin. Finally, hold the soldering-iron in the pad some seconds until the tin is touching the pad. When remove the soldering-iron from the pad, do not blow. And it is very important that every time a soldering is done, the end of the soldering-iron must be cleaned. The things shown in the middle of the next figure are used to clean the soldering-iron.

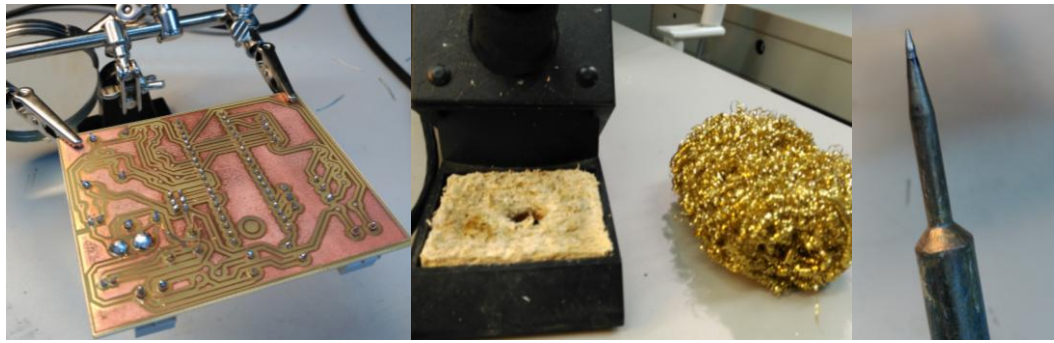


Figure 22: Soldering.

Another tip to soldering is to leave a small gap between the board and some components such as resistors or diodes, so that the air can flow through this gap to dissipating heat.

Note that this project has used sockets to connect the Microcontroller and the HC - 05 Module Bluetooth, in order to keep them portable and always be able to take them out easily from the embedded system. Also, this project uses two sockets of two pins in order to connect the electric strike and the indoor button so that both parts can be far away from the embedded system. Another remark is that the user has to be aware how to connect the components, for example, LEDs has a longer terminal to show the anode, or which pin of the three that the transistor has is the base, collector and emitter, or the polarity of the diodes and capacitors.

After soldering all the components, the prototype is ready to be tested.

5 SYSTEM TESTS AND RESULTS

This chapter explains all the methods that this project has performed in order to test the door lock system and describe the achieved results of the project.

Before manufacturing the PCB, the system had to be tested to check if it could work properly or not. The first tests were made building the designed circuit in a breadboard. A lot of tests have been done in the breadboard, first when the system did not include the indoor button to open the door nor the power supply circuit with the IC7805 component. Then it was thought that an indoor push button was absolutely necessary to the project, so it was tested again with this part. And finally, the project was tested with the design of the special power supply circuit. When everything worked in the correct way, a first PCB was developed. When it was tested, it did not work because of small distances between the traces and the copper set in PADS, however when soldering the components, it was easy to make a short in some connections. Then a second PCB was created, and when it was tested some smoke came out in some components: the pins of the DC barrel power jack were connected in the incorrect way; that is why the user must be careful when making the connection in PADS. Finally, after changing the PADS prototype, the third PCB was developed. To avoid the problem that happened in the first PCB, the distances between traces and between trace and copper were increased.

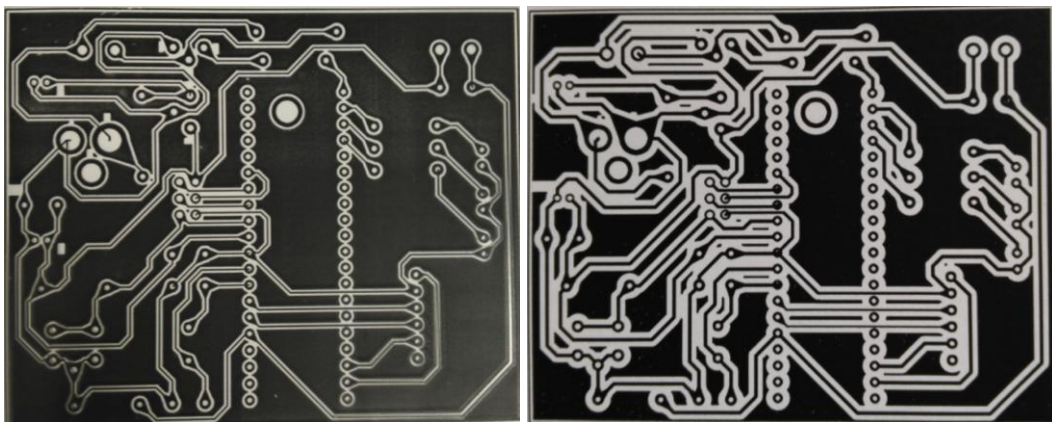


Figure 23: Printed circuits to manufacture the PCB.

The picture on the left is the circuit that was printed on the board to perform the first PCB. As the image shows, the width traces and the distance between trace and copper are too small therefore, the lines could be easily destroyed and make a short while soldering. By contrast, the figure on the right side shows the last printed circuit implemented; all the widths have been increased to avoid mistakes during the manufacturing process. The third PCB created was tested and worked correctly; later it is shown and explained in further detail.

This was the equipment needed along all the tests:

- AVR Universal Board.
- Some breadboards for placing the components and wiring.
- Cables and all the components shown in the previous images.
- Electric strike.
- Tools such as screwdriver to connect the cables to the sockets and to the electric strike.
- ATmega32 microcontroller.
- HC – 05 Bluetooth Module.
- AC/DC adapter to provide the 12V DC power supply.
- AVRISP mkII.
- Smartphone with the Android App installed.

Note that the software part had to be configured to the HC – 05 Bluetooth Module the first time before using it with the AT commands explained briefly in the software structure chapter of this report. Also, say that it is impossible to connect the SPI port to the breadboard, so when the microcontroller was placed in the breadboard, the cables were connected directly from the microcontroller pins to the AVRISP mkII (difficult way), allowing the microcontroller to be programmed. As mentioned in the previous list, an AVR Universal Board was used to test the system making it easy to interface with the microcontroller. If the ATmega32 is placed in this Universal Board, the user can use the SPI port provided by the board (easy way).

This was how the system looked in the breadboard:

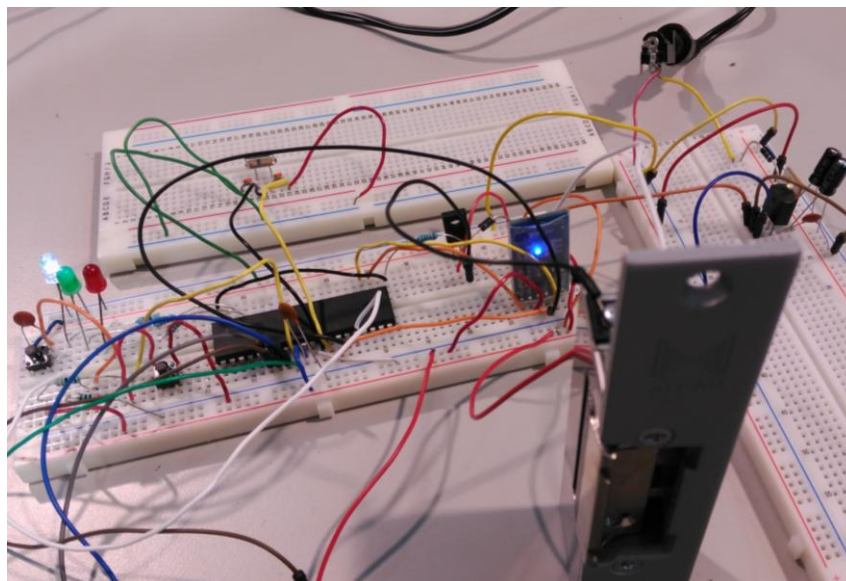


Figure 24: System test in the breadboard.

This picture was taken during the last system test on the breadboard, because the system includes the power supply with the IC7805 (placed in the breadboard on the right side of the picture), and the “indoor push button” with the capacitor for debouncing (shown on the left side of the picture). The button used here is not the final component; it is just a simulation to test the indoor push button to prove it works. As shown in the image, to test the system in the breadboard it was necessary to place all the connections through wires, making it too messy and difficult to understand, hence not user-friendly. Lastly, while wiring it was very important to check all the connections carefully to avoid mistakes.

The tests performed with the breadboard were successful and the system worked as expected. The Smartphone connected to the Bluetooth Module and opened the door. Thus, the PCB was manufactured.

When the PCB is ready to test, the electric lock and the indoor push button had to be connected to the system to supply the system with 12V DC. Plugged in, the working LED (white LED) was turned on to tell the user that the ATmega32 works, and the blue LED in the HC-05 blank fast, which means that normal mode worked (not AT mode) and no users connected to it. After, the user connects to the device, the LED is supposed to blink slowly. As the “open button” on the An-

droid app is pushed, the green LED switches on and the electric strike unlocks the door for 4 seconds, allowing the user to enter. Vice versa with the indoor button. The results have been successful.

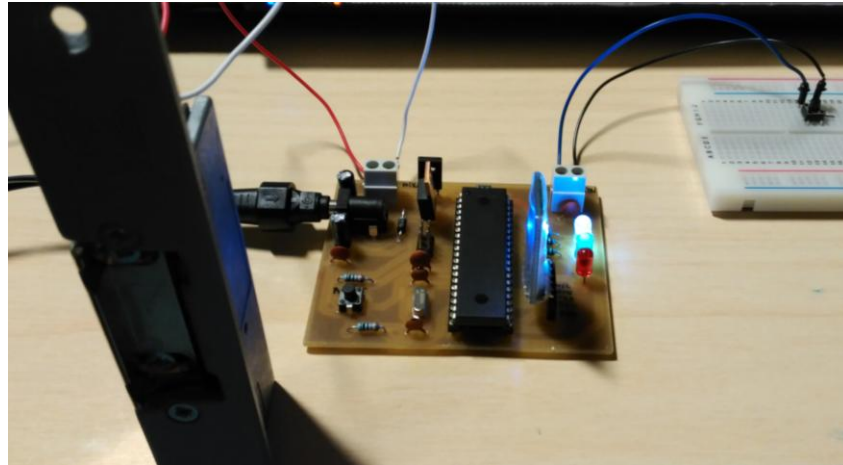


Figure 25: Smart Lock System prototype working.

When the final PCB system was tested, it accomplished the purpose of the project.

The only problem of the prototype was the Android App was not able to fetch the feedback from the microcontroller, however when tested with the Universal Board, the Smartphone app got the signal feedback. The difference between the project prototype and the Universal Board is the system oscillator. This might be

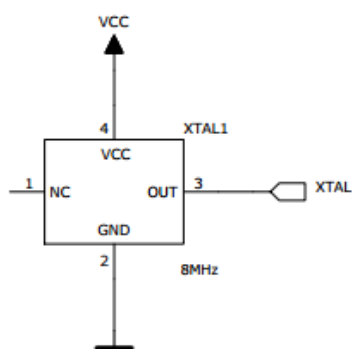


Figure 26: Universal Board Oscillator.

the reason for the problem. The universal board provides a compact full oscillator, which works as the system clock, unlike the project system where only a quartz crystal is used. This could have made the clock system for the prototype unstable and lead to failure. The next prototypes of the system should include a full oscillator and be tested to know the real problem. In the future expansion of the system the user could get the feedback messages in the Android app.

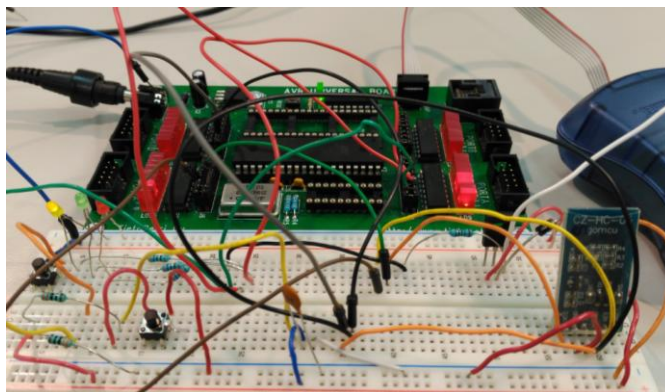


Figure 27: AVR Universal Board system test.

With the AVR Universal Board, it is possible to use its own oscillator, which inside has the crystal and all the electronics needed to make the signal conditioning such as an amplifier, avoiding using the system crystal oscillator. There is another small difference between this board and the project prototype, which is referred to the microcontroller AVCC pin to supply voltage pin for Port A (and the A/D converter). In the prototype this pin connected to nowhere, but in the datasheet of the microcontroller it is said that this pin should be connected to the VCC even if the ADC is not used. In the case of use the pin goes through a low pass filter to the VCC, as the AVR Universal Board is connected.

Some remarks:

- As mentioned before, the oscillator does not work as good as expected. Future prototypes should include the one shown before.
- In the door strike part, it was said that the SB130 diode was not a good choice to place to parallel with the electric strike, because its maximum DC blocking voltage was only 30V, which is too little to protect the BD139 transistor. Consequently its temperature increased a lot when giving power to the electric strike. Then, SB130 diode was unsoldered and a 1N4004 was placed instead, which has a maximum DC blocking voltage of 400V. Still the transistor temperature was increasing considerably when supplying the electric strike. For future researches, study this affair.
- When the IC7805 is connected longer to the power, its temperature increases a bit, because the input voltage is 12V and the output is 5V, so this

component has to dissipate a lot of energy. The same as before: for future investigations, study this issue.

Finally, some calculations of the system:

- When the electric strike is powered, with the configuration of the system: $I = 0.5A$, $V =$ around $5V$ (calculated with a tester). Because it has a solenoid inside, the voltage drop between its pins starts at $4.8V$ and it keeps increasing during the time that the electric strike is supplied. The last value of the voltage, after the 4 seconds, can be $0.4V$ more. And if the user doesn't let some time so that the electric strike can "discharge", its initial drop voltage will be more than $4.8V$, maybe $5.2V$, and after the 4 seconds, it will be, for example, $5.6V$.
- If the drop voltage in the door strike is $5V$, showing the configuration of this part of the circuit, the voltage drop between the collector and the emitter of the transistor will be $7V$ (because the supply is $12V$), and measuring it with the tester, is like this. And if the electric strike stays locked, the V_{CBO} in the transistor is $12V$. But there is nothing to worry about because the $V_{CBO\ max}$ is $80V$.
- Measuring with the tester, the output of the microcontroller is not exactly $5V$; it is a bit more than $4V$. There must be power loss in some point because theoretically, the microcontroller output voltage should be $5V$ (V_{CC}), the voltage, which the microcontroller is connected to.

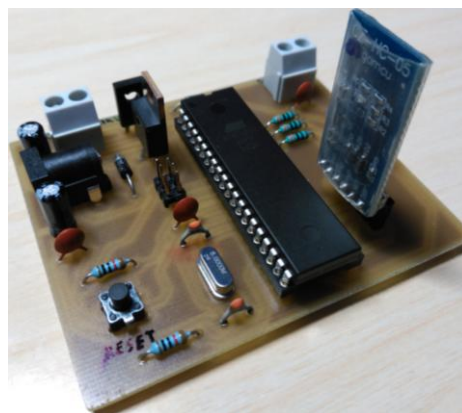


Figure 28: Result of the hardware part.

6 CONCLUSIONS AND IMPROVEMENTS

The Smart Lock System implemented in this thesis achieved the goals set successfully. It could control the access to a room allowing authorized users to access the area while using an Android App. In order to have a realistic test, the embedded system of the smart lock is mounted on a door prototype. The objective is accomplished by the teamwork of the two parts of the project: the software and the hardware, through all the research in different fields, such as embedded systems design, PCB manufacturing and software design. The prototype has been created so that the client gets a simple and small device, which can be implemented easily at any door, without changing the main structure. The main idea of this report, behind the design and implementation of the prototype, is to create a completely and helpful guide for future students who want to continue this project, and fortunately, this objective has also been achieved.

In particular, the hardware solutions designed in this project have been successfully implemented: the microcontroller can send a signal to the door strike allowing it to open the door when it gets the signal from the Bluetooth module or when it gets a signal from the indoor button; the microcontroller can be programmed through the SPI port implemented in the board for future configurations of the program; the system is correctly supplied through two different circuits; there is a working reset button in case of microcontroller failure; the user will know about the state of the door because of the LEDs; the embedded system is as small as possible.

The final goal of this project is to create a home automation system, which can control all the parameters and sensors inside a user home, such as the temperature, the illumination, etc. This project work achieved to implement the door lock system for this home automation. A further development of this work is the integration of home automation system in order to improve and add other parts that normal users would need or require to remotely control their home using their own Smartphone. Some ideas: making the door open when the user is in front of the door without having to use the Smartphone and pushing the button (auto connect-

ing to the Bluetooth and check if the information is valid), or adding new ideas like a monitoring task. In this case, a monitoring task would be really interesting choice to install in Technobothnia, because then, the staff in the Info room could know who is inside the rooms from his working place. Another idea is to place an external battery in case of power loss; this could be done with a demultiplexer.

The technology used in this project was Bluetooth since nowadays there is much information about it and a lot of devices that support Bluetooth. However, NFC technology is getting more and more popular, and it would be considering choice to the smart lock system. At the beginning it was thought to add this part also to the system through an OR gate, which could get both input signals, but because of the lack of time to research about this new technology, it was not possible.

In the results chapter, there are some tasks that the next students who continue with this work have to take care of. For example, the oscillator used in this project is a quartz crystal, and it does not work as good as expected. But the oscillator implemented by the AVR Universal Board works in the correct way, therefore this oscillator should replace the quartz crystal. Also, the temperatures of the BD139 transistor and the IC7805 increase a bit. First, the temperature of the BD139 transistor is increasing, even changing the SB130 diode to the 1N4004, because of the solenoid inside the electric strike, so here there is a research work to find out some techniques or methods to avoid this problem. As for the IC7805, which has to dissipate a lot of energy (from 12 to 5V), it is the same.

Finally, it is important to remark that this project gives a wide insight in the field of electronic engineering, because it has a lot of electronic concepts involved, and helps to familiarize with the development process of a real project.

REFERENCES

Printed books:

Allan R. Hambley (2012). *Electrónica*, 2ª Edición. Pearson. Prentice Hall.

Electronic publications:

Mentor Graphics (2016). PADS – Resources. Source: <https://www.pads.com/>

ALCAD (2010). 201 Line. Door Entry Systems. Technical Manual. Source: www.alcad.net

PhaseLink Corporation (2005). Pb-Free and Green Package. Source: http://www.phaselink.com/QA/Green_package.pdf

Datasheets:

ATmega32(L) Datasheet (2001). Atmel Corporation. USA.
<http://www.atmel.com/images/doc2503.pdf>

HC – 05 Bluetooth Module Datasheet.
[http://wiki.iteadstudio.com/Serial_Port_Bluetooth_Module_\(Master/Slave\)_:HC-05](http://wiki.iteadstudio.com/Serial_Port_Bluetooth_Module_(Master/Slave)_:HC-05)

BD139 Datasheet (2007). Fairchild Semiconductor Corporation.

<https://www.fairchildsemi.com/datasheets/BD/BD135.pdf>

SB130 Datasheet (2003). Diodes Incorporated.

http://www.diodes.com/_files/datasheets/ds23022.pdf

1N4001 and 1N4004 Datasheets (2014). Diodes Incorporated.

http://www.diodes.com/_files/datasheets/ds28002.pdf

7805IC Datasheet (2001). Fairchild Semiconductor Corporation.

<http://www.engineersgarage.com/sites/default/files/7805.pdf>

Internet URL sources:

Microcontrollers guide:

<http://www.atmel.com/products/microcontrollers/default.aspx?src=parent>

Bluetooth and HC - 05 guides and explanations:

<https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics>

<https://arduino-info.wikispaces.com/BlueTooth-HC05-HC06-Modules-How-To>

<http://www.geekfactory.mx/tutoriales/bluetooth-hc-05-y-hc-06-tutorial-de-configuracion/>

<http://www.electronica60norte.com/mwfls/pdf/newBluetooth.pdf>

Transistor as a switch:

<http://www.allaboutcircuits.com/textbook/semiconductors/chpt-4/transistor-switch-bjt/>

http://www.electronics-tutorials.ws/transistor/tran_4.html

<http://www.educachip.com/transistor-bjt-como-interruptor/>

Choosing the transistor and the resistor in its base terminal:

<http://electronicsclub.info/transistors.htm#choosing>

<https://www.sistemasorp.es/2011/10/05/calcular-la-resistencia-para-un-transistor-accionado-por-un-microcontrolador/>

Door strike:

http://www.dipolnet.com/electric_door_strike_abr-001_for_alcad_door_entry_system_S973000.htm

http://www.alcad.net/uploads/publicaciones/catalogos/pdf/ManTecP_I.pdf

<http://www.locksmithledger.com/article/10238119/electric-strikes-everything-you-should-know>

Diode in parallel with electric lock:

<http://www.iluselectronics.com/q-why-is-it-essential-to-connect-a-diode-in-reverse-polarity-in-parallel-to-a-dc-operated-electric-lock/>

7805 Voltage Regulator:

<http://www.rakeshmondal.info/IC-7805-Voltage-Regulator>

<http://www.electronicshub.org/understanding-7805-ic-voltage-regulator/>

Debouncing:

<http://whatis.techtarget.com/definition/debouncing>

<https://www.newbiehack.com/UnderstandingButtonDebouncing.aspx>

Pull-up and pull-down resistors:

<https://learn.sparkfun.com/tutorials/pull-up-resistors>

SPI Port:

<https://www.arduino.cc/en/Reference/SPI>

<https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>

Oscillator:

<http://electronics.stackexchange.com/questions/160921/when-we-need-external-crystal-for-atmega>

<http://100ciaencasa.blogspot.fi/2015/01/tutorial-electronica-basica-14.html>

Vacuum UV Exposing Device:

<http://www.printer-manufacture.com/Precise-Vacuum-UV-Exposure-Unit-p880.html>

AVR Universal Board:

<http://www.tietopetri.fi/tuotteet.html#uni>

