WIRELESS SYSTEM OF ACTUATION AND SUPERVISION IN PNEUMATIC ACTUATORS THROUGH ZIGBEE PROTOCOL AND ARDUINO

Adauto Granja
Marcelo Henrique de Almeida Orlandin
Lucas Emanuel Soeira
Universidade Paulista UNIP – Av: Carlos Consoni, 10 – Jardim Canadá – Ribeirão Preto - SP
adautojunior2014@gmail.com
marcelo.orlandin@hotmail.com
soeira@sc.usp.br

Abstract. *This paper presents an actuation and supervision system of pneumatic actuators through ZigBee protocol that enables wireless communication by radiofrequency between these actuators and the microcontroller Arduino. The graphical interface of actuation and supervision is programed in Visual Basic Language, and is responsible for sending commands via serial port to a module called ZigBee Coordinator. This module transmits the information to the Arduino, and monitors the sensor connected to it via ZigBee modules called end devices. Arduino receives information from the ZigBee coordinator, interpreting them according to its program, developed in its platform, and acting on the actuators and other instruments that may be incorporated into the system, such as sensors and transmitters. Twenty-four volt digital sensors with three wires, containing 50-mm optical sensitivity and 5-mm capacitive sensitivity, are used. The use of wireless communication was opted along with Arduino because of various advantages offered by this system, such as: economy of signal cabling, ease of maintenance and energy savings. Besides these factors, it is important to emphasize that this system operates on an open platform, allowing the creation of low-cost sensor devices, thus providing a wide variety of instruments connected to the same network.*

Keywords: *Arduino, ZigBee, Pneumatic actuators, supervision, radiofrequency*

1. INTRODUCTION

The increasing need for efficient and autonomous systems, coupled with the advancing technologies of control and communication solutions, makes low cost microcontroller accessibility and network trustworthiness necessary. This tendency is increasingly more common in automated systems applications (Severino, 2008).

In this scenario, wireless network communication is growing rapidly due to the advantages offered by them, such as economy of signal cabling, low cost installation, and ease in planning and installation (Lee *et al.*, 2007).

Within this technology of wireless communication, there is ZigBee protocol, from ZigBee Alliance. ZigBee, which is still under development, is based on IEEE 802.15.4 standard protocol, and allows operation in the following frequency bands: 2.4 GHz, 915 MHz and 868 MHz. These are free and do not require a license for users, which is making ZigBee a comprehensive global protocol.

Observing this tendency, this work aims to integrate this protocol into the Arduino microcontroller, through a supervision system, and operates the electronic actuators. Every process can be followed in a graphical user interface developed using Visual Basic, which will display status information from the sensors and buttons to drive the operating process.

The Visual Basic program is responsible for sending commands, via serial port, to a module called ZigBee coordinator. This module transmits the information to Arduino, and monitors the sensor connected to it, via ZigBee modules called end devices. Arduino receives information from the ZigBee coordinator, interpreting them according to its program, developed in its platform, and acting on actuators.

2. MATERIALS AND METHODS

2.1 Arduino

The electronic board, called Arduino, was created in 2005 by a team of Italians with the intention to create a micro controller with open source hardware and software, and that was affordable to the majority of people (Arduino, 2013a).

Because it is an open platform, Arduino brings a great advantage to users: a commitment that volunteers take to work to improve software and hardware. In addition, there is interaction between several companies that are selling kits for Arduino (Info. Brazil, 2009).

Thus, this work platform has grown a lot since its creation, becoming applicable to various types of situations. The Modelix Company, a manufacturer of robotic kits, has been incorporating this control board into its products, such as application control lift (Leomar, 2013). Other examples of applications are haptic glove (Pfleger, 2012.) robotic arms, speed meter (Nunez, 2012).

The Arduino board is able to work in various types of automation that is necessary to program outputs according to the variation of the input signals. Importantly, this adapter also works with communication protocols such as Bluetooth, Internet, ZigBee and Modbus. Figure 1 illustrates an Arduino Mega 2560 plate.

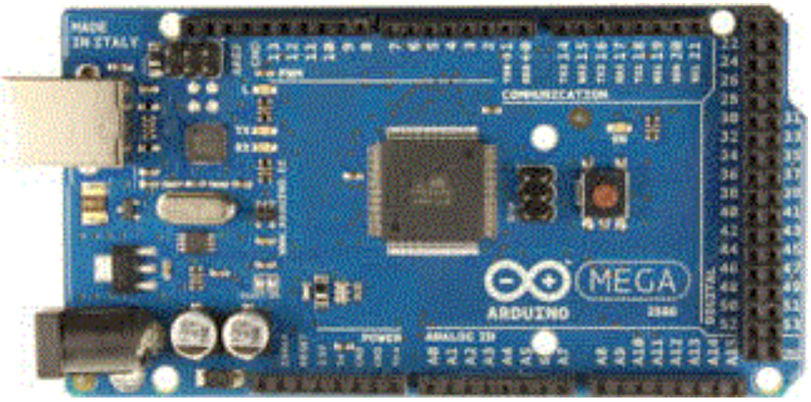


Figure 1 – Arduino ATMega 2560 board

2.1.1 Arduino’s Hardware

The first board created by Arduino's team was Arduino Uno, but over the years, this model has undergone many changes, and today Arduino has several different models: Arduino Due, Arduino Explore, Arduino Mega ADK, Arduino BT (Arduino, 2013b).

Table 1 compares some of the microprocessors that compose an Arduino board and its properties (ATmega, 2006).

Micro processor	FLASH	EEPROM	RAM	I/O	16 bits de resolution	Serial UARTs	Channel ADC
Atmega640	64KB	4KB	8KB	86	12	4	16
ATmega1280	128KB	4KB	8KB	86	12	4	16
Atmega1281	128KB	4KB	8KB	54	6	2	8
Atmega2560	256KB	4KB	8KB	86	12	4	16
Atmega2561	256KB	4KB	8KB	54	6	2	8

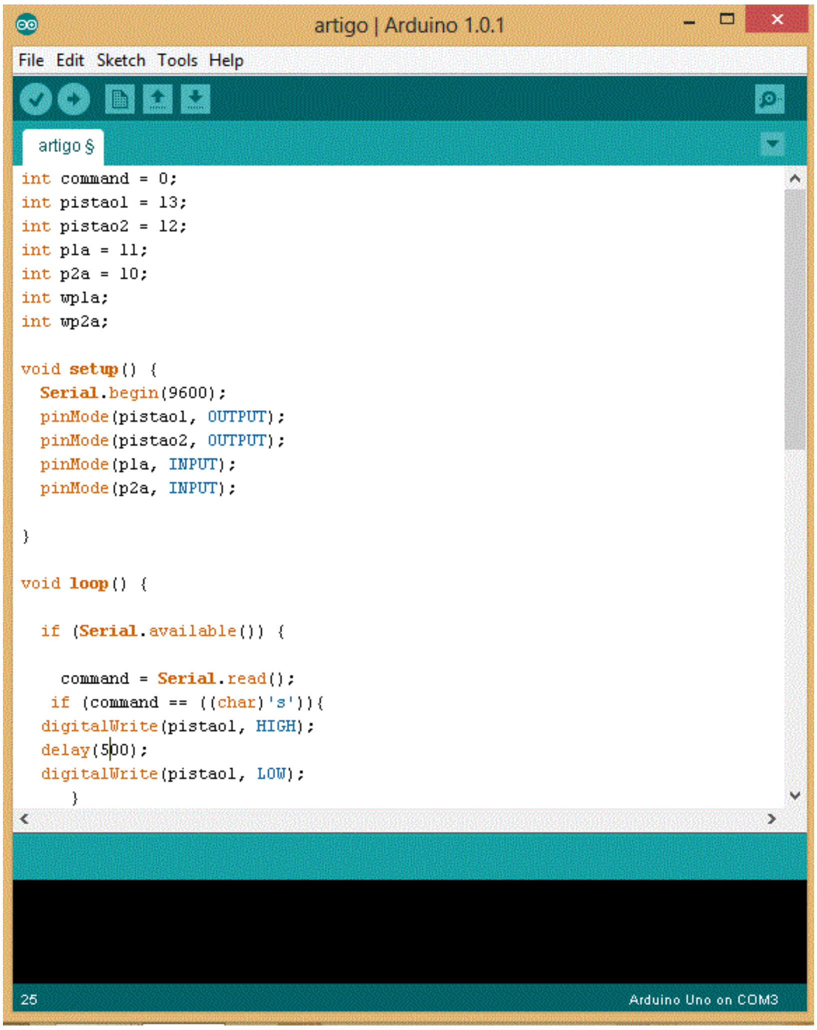
Table 1 – Arduino board and its properties

The signboard used in this work will be the Arduino Mega, with a microprocessor based on ATmega 2560. It has 54 digital input / output pins, 16 analog inputs, 4 UARTs (hardware serial ports), a crystal oscillator of 16 MHz, a USB connection, an input power, an ICSP connection, and a reset button (Arduino, 2013c).

2.1.2 Arduino’s Software

The developmental environment is a multiplatform application, developed in Java, with an open source code, which runs on Windows, Linux and Mac (Werneck, 2009). Thanks to the input by multiple users, libraries with more functions are being incorporated into the software, which makes development projects more dynamic and broad.

The function of this IDE is beyond providing a simple programming environment; it compiles the code and downloads this file into the microcontroller via a USB cable. Besides being a programming environment and compiler, it also has a communication box able to send and receive serial information**.** Figure 2 portrays the variable process declaration; it also illustrates how output driving is made from a serial port reading. (Gibb, 2010).



```
int command = 0;
int pistao1 = 13;
int pistao2 = 12;
int p1a = 11;
int p2a = 10;
int wp1a;
int wp2a;

void setup() {
  Serial.begin(9600);
  pinMode(pistao1, OUTPUT);
  pinMode(pistao2, OUTPUT);
  pinMode(p1a, INPUT);
  pinMode(p2a, INPUT);

}

void loop() {

  if (Serial.available()) {

    command = Serial.read();
   if (command == ((char)'s')){
  digitalWrite(pistao1, HIGH);
  delay(500);
  digitalWrite(pistao1, LOW);
     }
```

Figure 2 – Arduino IDE

2.1.3 Arduino Shields

Several different manufacturers make the shields for Arduino boards, whose service is to add extra functions to the Arduino. Examples: Ethernet, ZigBee, Bluetooth, Wi-Fi, EasyVr, among others (Labdegaragem, 2013). The study of ZigBee shield will be specified here, which will be the subject of study in this work. Figure 3 illustrates an Arduino board coupled to three shields, the last one being, ZigBee.

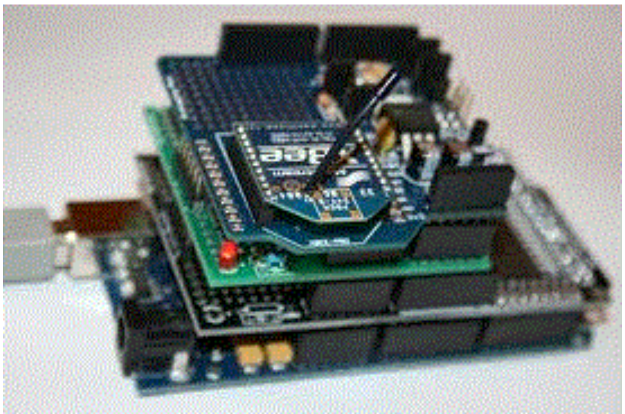


Figure 3 – Arduino board coupled to three shields

2.2 Zigbee Protocol

ZigBee is a communication protocol based on wireless technology that has been developed and perfected by Xbee Alliance. An alliance with large automation companies: Phillips, Schneider Electric, Texas Instruments, Legrand, among others (Zigbee, 2013a).

The ZigBee standard is based on IEEE 802.15.4 protocol and operates in the global 2.4 GHz frequency. It can also operate on frequencies of 915 Mhz in the Americas and 868 MHz in Europe. A great advantage it has in relation to other wireless technologies is that it is not necessary to acquire a license to use it.

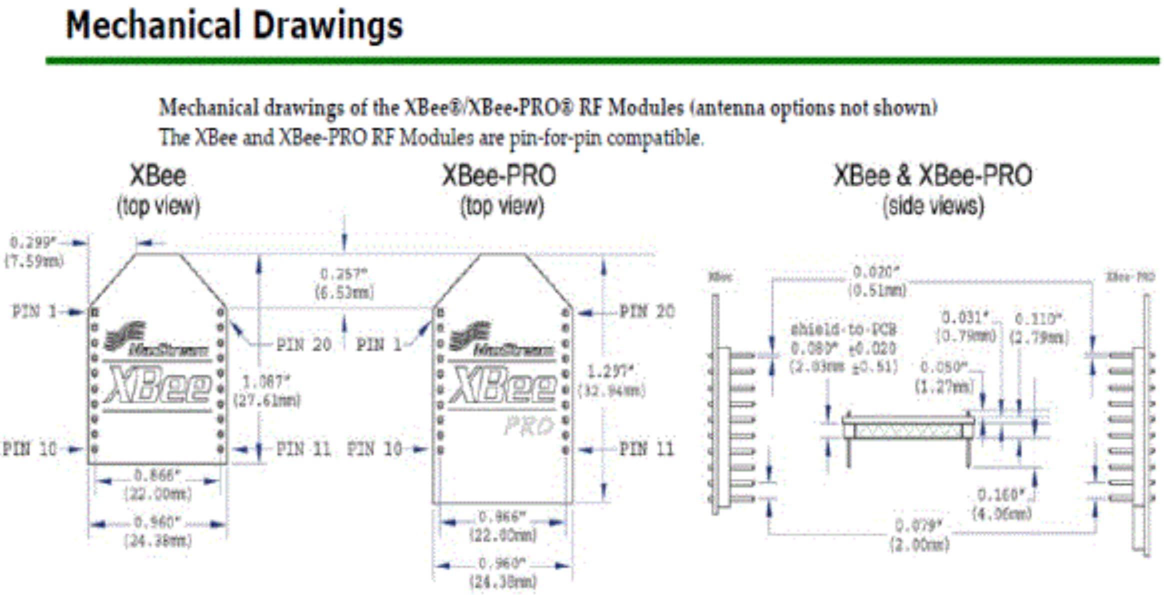
Only ten years on the market, the ZigBee standard today is widely used in building and residential automation. Although it has many advantages, this protocol is still debilitated as to the amount of data that can be transmitted through it, compared to other protocols already established in the industry, such as Fieldbus and Profibus (Lopes, 2012). Compared to these same protocols, ZigBee has the following advantages: low energy consumption, reduction of electromagnetic interference (since it is not necessary to use cabling), and the possibility to connect a much larger amount of equipment on the same network.

According to ZigBee Alliance, over 600 certified products currently use the ZigBee standard (Zigbee, 2013b). In order to develop this project, XBee radio frequency (RF) modules are used.

2.2.1 RF Xbee Module

The RF XBee module is currently manufactured by Digi and designed to meet the IEEE 802.15.4 standard, operating in the global frequency of 2.4 GHz (Xbee, 2013a). The RF XBee modules also come with digital inputs and outputs, analog inputs and PWM outputs (Xbee, 2013b). These inputs and outputs can be individually accessed through specific commands, called AT Commands.

Table 2 shows the general characteristics of the module used in this work. Table 3 shows the pin functions of the module. Figure 4 shows a dimensional drawing of the module. Figure 5 illustrates the structure of AT Commands (Xbee, 2013c).



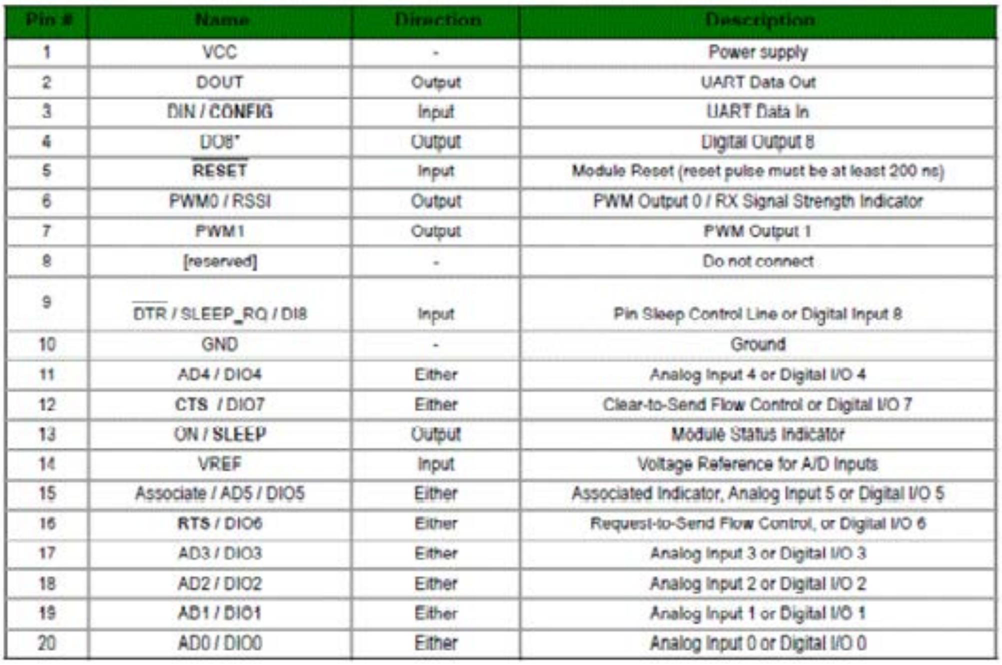
Mechanical Drawings

Mechanical drawings of the XBee®/XBee-PRO® RF Modules (antenna options not shown)
The XBee and XBee-PRO RF Modules are pin-for-pin compatible.

Figure 4 – X-bee Mechanical Drawing

Specification	*XBee*
Performance	
Indoor/Urban range	*Up to 100 ft (30m)*
Outdoor RF line-of-sight Range	*Up to 300 ft (90m)*
Transmit Power Output (software selectable)	*1mW (0 dBm)*
RF data rate	*250Kbps*
Serial Interface Data Rate	*1200bps - 250Kbps (non-standard baud rates also supported*
Receiver Sensitivity	*-92 dBm (1% packet error rate)*
Power Requirements	
Supply Voltage	*2.8V - 3.4V*
Transmit Current (typical)	*45mA (@3.3V)*
Idle/Receive Current	*50mA (@3.3V)*
Power-down Current	*<10µA*
General	
Operating Frequency	*ISM 2.4Ghz*
Dimensions	*0.960" x 1.087" (2.438cm x 2.761cm)*
Operating Temperature	*-40 to 85° C (Industrial)*
Antenna Options	*Integrated Whip Antenna, Embedded PCB Antenna U.FL Connector, RPSMA connector*
Networking & Security	
Supported Network Topologies	*Point-to-point, Point-to-multipoint & Peer-to-peer*
Number of Channels (Software selectable)	*16 Direct Sequence Channels*
Addressing Options	*PAN ID, Channel and Addresses*
Agency Approvals	
United States (FCC Part 15.247)	*OUR-XBEE*
Industry Canada (IC)	*4214A XBEE*
Europe (CE)	*ETSI*
Japan	*R201WW07215214*
Australia	*C-Tick*

Table 2 - General characteristics of the module



Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / $\overline{\text{CONFIG}}$	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	$\overline{\text{RESET}}$	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	$\overline{\text{DTR}}$ / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / DIO6	Either	Request-to-Send Flow Control, or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

* Function is not supported at the time of this release

Table 3 – Pin Assignments for the Xbee and Xbee-PRO Modules

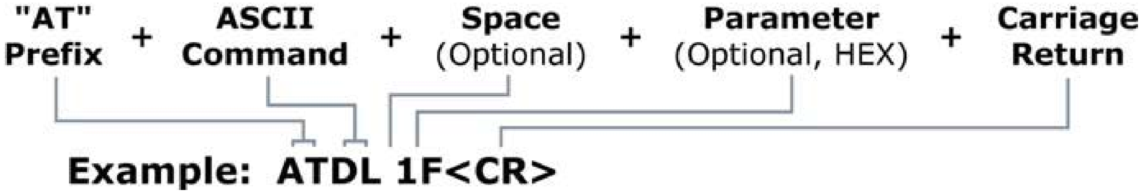


Figure 5 - Structure of AT Commands

Another way to make communication is through direct access to the module pins that are on the network. Thus, digital or analog output of a module is directly related to the input of another one. This is the approach used in this work.
The Xbee programmer is used to configure the network that communicates serially with the computer. This programmer sends AT commands through a program called X-CTU to the modules, configuring them. Table 4 illustrates the AT commands used for the creation of the network.

AT Commands		
+++		Enables programming
ATID	PAN ID	Defines an ID for modules network communication (PAN ID). All modules of the network must have the same one.
ATMY	My Address	Sets the module address on the network
ATDn	I/O pin configuration	Defines the settings of I/O pins. “n” is the number of digital input or output.
ATIA	I/O input address	Defines the address of the transmitter module.
ATDL	Destination address low	Defines the address of the receiver module.
ATIR	Sample Rate	Sets the monitoring frequency of digital gates.
ATPR	Pull-up-Resistors	Disables internal pull up resistors of digital input pins.
ATWR	Write	Saves changed settings

Table 4 - AT commands and their definitions

Two modules are used for communication: transmitter and receiver. Table 5 shows settings of the transmitter module. Table 6 portrays receiver settings.

Radio Transmitter		
Reset	ATRE N/A	resets the radio to its factory settings
PAN ID	ATDID 3001	any address from 0 to FFFE will do
MY Address	ATMY 1	
Destination address high	ATDH 0	indicates a 16-bit address
Destination address low	ATDL 2	the address of the receiving radio
PIN 0 I/O configuration	ATD0 3	digital input
Sample rate	ATIR 14	20ms, 14 in hexadecimal = 20 in decimal
Pull-up resistors	ATPR 0	disables internal pull-up resistor on all pins
Write to memory	ATWR N/A	saves the settings to flash memory

Table 5 - Transmitter Module Settings

Radio Receptor		
Reset	ATRE N/A	resets the radio to its factory settings
PAN ID	ATID 3001	any address from 0 to FFFE will do
MY Address	ATMY 2	
Pin 0 I/O configuration	ATD0 5	digital output, HIGH on startup
I/O Input Address	ATIA 1	address of the transmitting radio
Write to Memory	ATWR N/A	saves the settings to flash memory

Table 6 - Receiver module settings

Figure 6 illustrates electronic diagram transmission signal, which is triggered by a digital output from Arduino. Figure 7 shows a diagram of signal receiving.

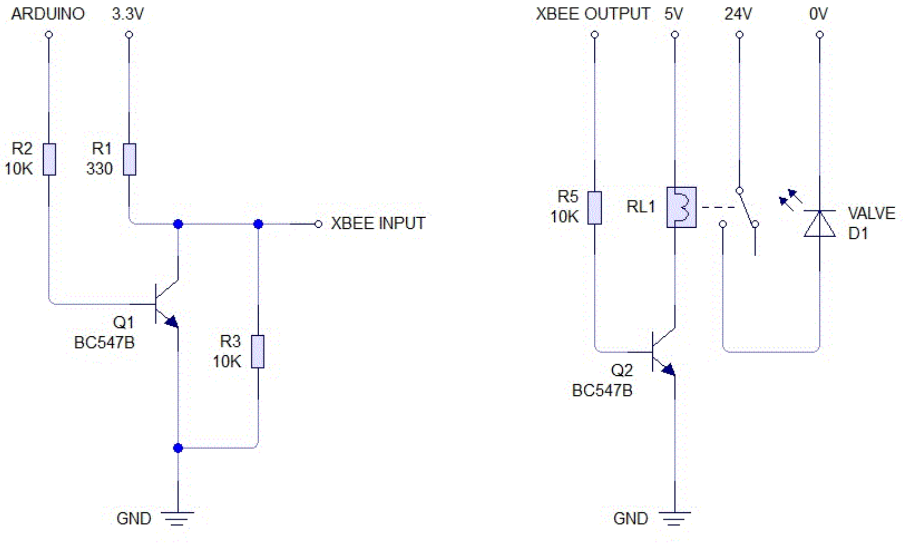


Figure 6 - Diagram of a sending signal from a Xbee Module

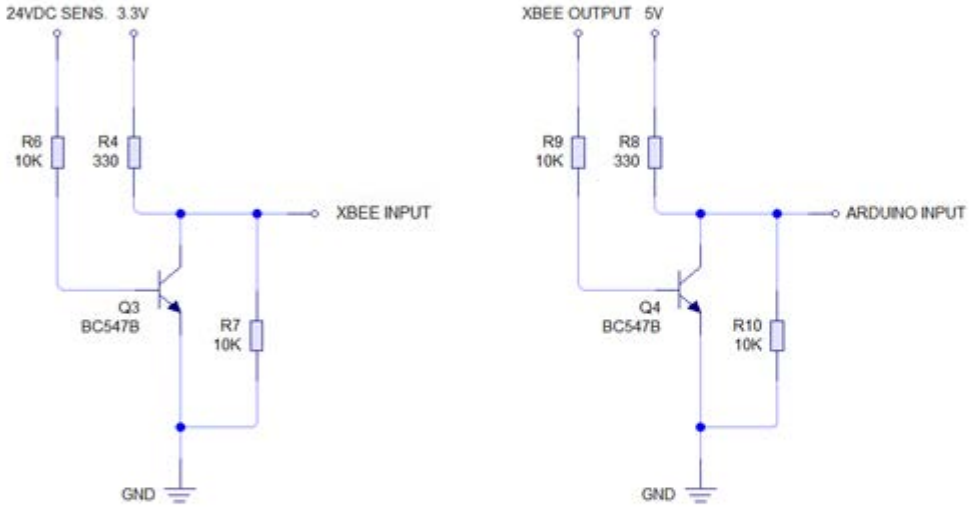


Figure 7 - Sensor signal reception

2.3 Visual Basic

Visual Basic is a programming language developed by the Microsoft Company, initially designed to start people in programming. At first, the language was simpler, named only Basic, but a succession of changes and improvements made this language become Visual Basic, as we know it today (Roman *et al.*, 2002; Cornel and Morrison, 2002).

Visual Basic is a tool for object-oriented programming that allows the creation of graphical user interfaces, database access, serial communication, etc. Visual Basic was chosen as the platform for this work because it has simple programming language, a dynamic graphical interface, and has features that facilitate communication through the serial port between the program developed and Arduino.

2.4 Simulation

This work will drive two pneumatic cylinders that simulate the rejection, or not, of a piece according to its size. The system rejection, which due to the didactical plant of the Festo Company, provided by the Paulista University, not using a treadmill, will be tested on a workbench. A support for the inductive sensor is manually placed so that the piece will pass through it. If the piece is the right size, nothing happens. However, if the piece triggers the sensor, this will initiate a pre-programmed sequence of activation in pneumatic cylinders located in the Festo plant. Figure 8 shows pneumatic diagram of the system. Figure 9 illustrates electronic actuation diagram of the valves. Figures 10 is a portrait of the pneumatic circuit mounted on the lab at Paulista University.

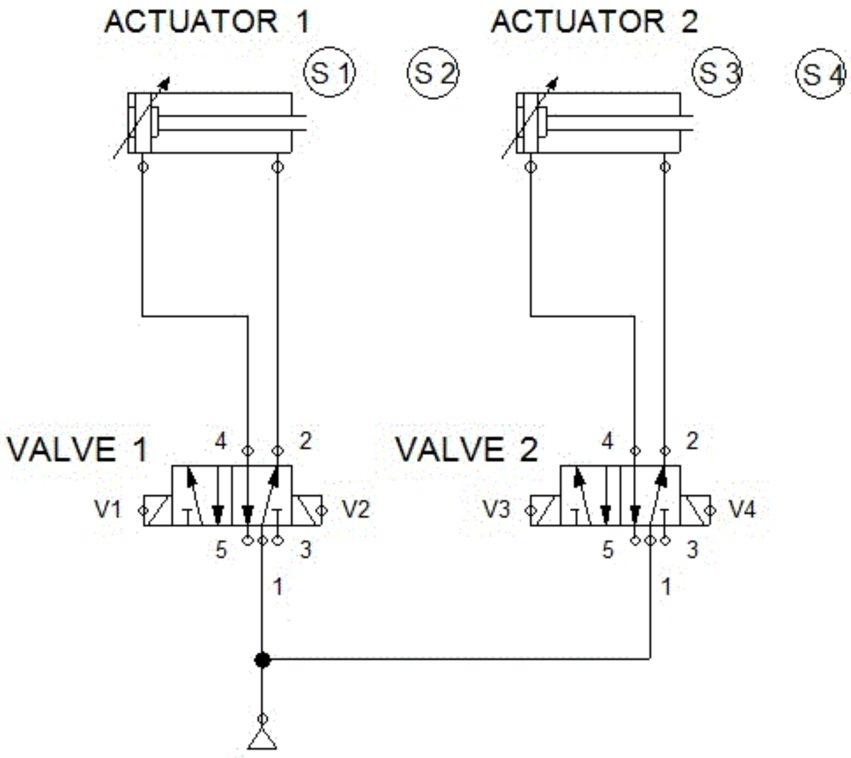


Figure 8 – Pneumatic Diagram

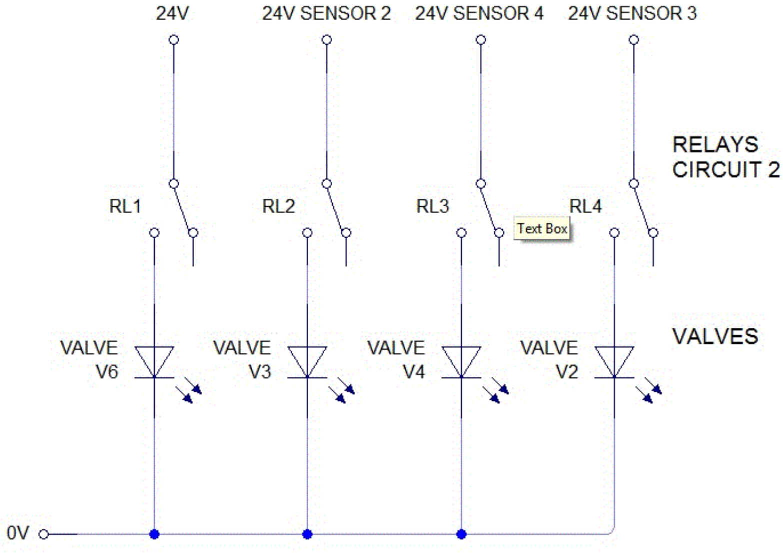


Figure 9

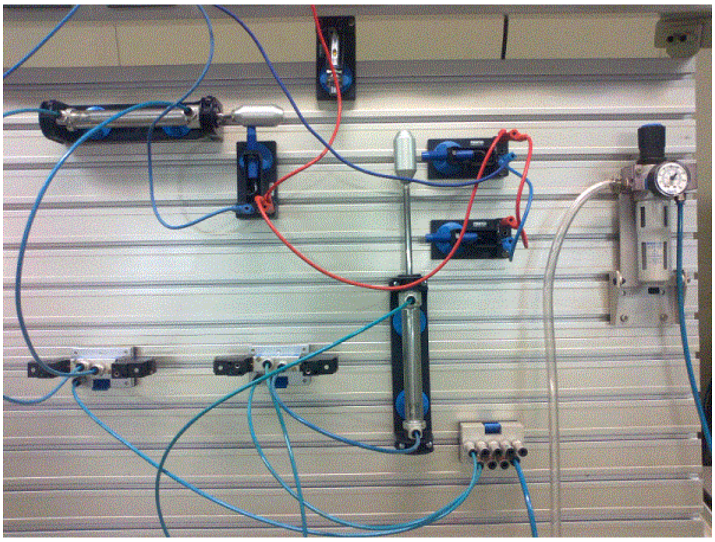


Figure 10 – Pneumatic Plant

3. RESULTS

For purposes of tests, 200 consecutive activations were conducted under the following conditions:

First condition: Less than ninety meters without physical barriers or signs of other frequencies.
Second condition: Less than ninety meters without physical barriers but with signals from other frequencies (frequency up to 1 MHz).
Third condition: More than ninety meters without physical barriers or signs of other frequencies.

Results:

First condition: the system presented an efficiency around 95%
Second condition: Even with signs of other frequencies, the system showed the same performance
Third condition: The system displays unsatisfactory performance

Observed during the workbench test of this system a satisfactory answer was shown as to ease programming and configuration of hardware and software involved. With regard to the response time of the commands, the system met expectations. The software used to develop the interface was efficient in sending commands, however, demonstrated limitations in the supervision.

4. CONCLUSION

XBee communication modules demonstrate some advantages, such as ease of installation and implementation, saving cabling and low power consumption of the RF modules. Allied to these advantages are the ease of programming, the large number of analog and digital inputs and outputs and variety of devices with which Arduino can work, make this system able to deploy in various types of application.

The software used to develop the GUI, although it worked, showed limitations in supervision, due to not being suitable for the development of supervisor interfaces. Thus, in future applications the use of a specific software development supervisor is suggested; such as Indusoft, iFix and ellipse because these, like the Arduino, have modbus communication drivers.

5. ACKNOWLEDGEMENTS

This project is supported by Paulista University.

6. REFERENCES

Arduino, 27 May. 2013a <http://arduino.cc/en/>

Arduino, 27 May. 2013b <http://arduino.cc/en/Main/Products?from=Main.Hardware>

Arduino, 27 May. 2013c < http://arduino.cc/en/Main/ArduinoBoardMega2560>

ATmega, Microcontroller with 640/1280/1281/2560/2561 Bytes In-System Programmable Flash, 2006

Cornell, Gary. and Morrison, Jonathan., 2002. *"Programming VB.NET: A guide for Experienced Programmers"*

Gibb, Alícia M. 2010. "New Media Art, Design, and the Arduino Microcontroller: A malleable tool". Feb. 2010

Info. abril, 2009. "O hardware em código aberto". 9 Mar. 2009 <http://info.abril.com.br/professional/tendencias/hardware-livre-leve-e-solto.shtml>

Labdegaragem, 27 May. 2013 <http://www.labdegaragem.org/loja/index.php/catalogsearch/result/?q=shield+arduino&x=-1115&y=-193>

Lee, J.S., Su, Y.W., Shen, C.C., 2007. *A comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee and Wi-fi*

Leomar, 27 May. 2013 <http://www.leomar.com.br/modelix/index.php?option=com_content&view=article&id=278%3A2108-2009-&catid=101%3Atv-modelix&Itemid=126&lang=pt >

Lopes, A.P., Nascimento, D.B., Lima, M.M.L. S., 2012. "Projeto de aplicação zigbee em rede de automação industrial".

Pfleger, S.G., 2012. "Plataforma Robótica Genérica Baseada em Arduino". 03 Mar. 2012 <http://www.das.ufsc.br/~raffo/SeminariosRobotica_DAS/11.Seminario_22.03.2012_S.G.Pfleger/Arduino%20e%20aplicacoes.pdf>.

Nunez, A.F., 2012. "Uma introdução ao Arduino e suas aplicações na robótica". 29 May. 2012 <http://pet.inf.ufpel.edu.br/sacomp/2012/palestras/SACOMP2012-29_05-4.Arduino.pdf>.

Roman, S., Petrusha, R., Lomax, P., 2002. "VB.Net Language, In a Nutshell . A desktop quick reference".

Severino, R.A.R.S., 2008. *On the use of 802.15.4/ZigBee for Time-Sensitive Wireless Sensor Network Application*, Instituo Superior de engenharia do porto, Oct. 2008

Werneck, P., 2009. "Introdução ao Arduino". 22 Jun.. 2009 < http://www.sabereletronica.com.br/secoes/leitura/1307>

Xbee, Product Manual, 13 Feb. 2013a "*v1.xEx - 802.15.4 Protocol For RF Module Part Numbers: XB24-A...-001, XBP24-A...-001*" pg.05

Xbee, Product Manual, 13 Feb. 2013b "*v1.xEx - 802.15.4 Protocol For RF Module Part Numbers: XB24-A...-001, XBP24-A...-001*" pg. 7

Xbee, Product Manual, 13 Feb. 2013c "*v1.xEx - 802.15.4 Protocol For RF Module Part Numbers: XB24-A...-001, XBP24-A...-001*" pg.27

Zigbee, 27 May. 2013a <www.zigbee.org/About/OurMembers.aspx>

Zigbee, 27 May, 2013b <http://www.zigbee.org/About/UnderstandingZigBee.aspx>

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.