

The conception and the implementation of a control system for servomotors with the application of a wireless network

Abstract. The aim of this study was to design and create a control system for servomotors with the application of a computer wireless network. To achieve this goal 802.11g wireless standard, servomotors, and open source Arduino platform were used.

Streszczenie. Celem niniejszej pracy było zaprojektowanie oraz wykonanie systemu sterowania serwo mechanizmami z wykorzystaniem bezprzewodowej sieci komputerowej w standardzie 802.11. Do osiągnięcia celu zastosowano standard 802.11g, serwo mechanizmy oraz open source platformę Arduino. (Koncepcja i implementacja systemu sterowania serwo mechanizmami z zastosowaniem bezprzewodowej sieci komputerowej).

Keywords: control system for servomotors, wireless network, 802.11, arduino.

Słowa kluczowe: układ sterowania serwo mechanizmami, sieć bezprzewodowa, 802.11, arduino.

doi:10.12915/pe.2014.02.36

Introduction

The world runs faster and faster everyday. New inventions appear in every corner of the world. This is a fact. But the reason cannot be more prosaic. Everybody wants to avoid physical activities. That is why intelligent buildings become more and more popular, cars park themselves or vacuum cleaners operate without our supervision. We simply do not know in how many different fields we are being replaced by machines. That is why this trend of conception of a control system for servomotors with the application of a wireless network is relevant. It can also be used in electrical engineering and electronics [1-9].

Many different solutions are already available on the market but our purpose was to present a solution that can be adopted not only in the industry or in everyday life but also by academics as interesting laboratory exercise that draws students' attention. To achieve this goal open source Arduino platform with the adaptation of wireless 802.11 standard [10] as the heart of the control system was chosen.

Conception

Every project begins with an idea. In this case the main idea was to control a servomotor via wireless IEEE 802.11 network [11]. To achieve this goal following establishment has been considered. Main schematic diagram is presented in figure 1.

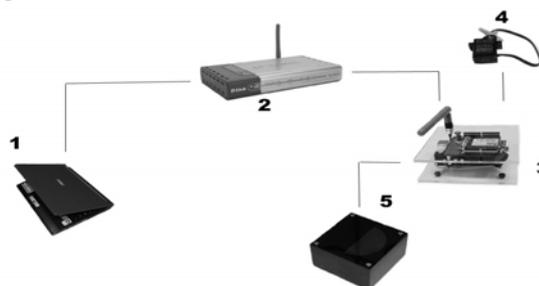


Fig. 1. Schematic diagram

1. Every PC contains a WCC (Wireless Control Client) application. For the purpose of the control system one was created with CodeGear Delphi programming language. Object language based on Pascal, this decision was dictated strictly by the experience of the programmer. In the future, creation of applications on Windows RT/Phone, Android, Symbian or iOS are possible.

2. As 802.11 protocol is basically used in small wireless networks either access point with DHCP server or ad-hoc connections between a computer and a control system was

required. In this case D-Link DSL-G684T wireless G54M ADSL2+ Modem Router w/4 Port 10/100 Switch capable to test connection WiFi-WiFi as well as Ethernet-WiFi.

3. A control system consists of Arduino Uno mainboard with additional WiFly SparkFun Shield (Tab. 1) [12], [13]. For this project the control system is called WCS (Wireless Control System).

4. We have chosen to control two simple 9 gram servomotors HXT900 by HEXATRONIK (Tab. 2). The main factors were low cost and relatively good construction.

5. As all new inventions without the suggestion of application are rather meaningless, to make this project a little bit more "alive" we decided to add one. To do so this "Black-Box" was created. Inside a small photocell and one servomotor are hidden. The purpose of this box is to maintain previously set light intensity inside. The top of the box works like a set of blinds in windows. Servomotor controls obscure the blinds while the photocell checks light intensity.

Table 1. Arduino specification

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14(6 PWM)
Analog Inputs Pins	6
DC Current per I/O Pin	40mA
DC Current for 3.3V Pin	50mA
Flash Memory	32KB(ATmega328),0.5KB used by bootloader
SRAM	2KB (ATmega328)
EEPROM	1KB (ATmega328)
Clock Speed	16MHz
Dimensions	2.7x2.1 inches (USB and power jack beyond)

Table 2. HXT900 specification

Modulation	Analog
Torque	4.8V: 1.6 kg-cm
Speed	4.8V: 0.12 sec/60°
Weight	9.1g
Dimensions	21x12x22mm
Motor Type	Coreless
Gear Type	Plastic
Rotation/Support	Brushing
Rotation Range	90°
Pulse Cycle	20ms
Pulse Width	450-2450 μs
Connector Type	JR

Arduino platform capabilities are also presented in [15], [16], [17].

Application

The main issues of new inventions are the applications. History of man kind is full of gadgets created without purpose that land on shelves never to be used again. This is why few examples of implementation of a control system for servomotors via wireless network are presented.

STAGE LIGHT CONTROL: separately controlled stage lights with an ability of preprogrammed movement scenarios or manually controlled. The only limitation in the number of lights is determined by the quality of intermediary devices forming a wireless network (with an emphasis on DHCP server).

SPOTLIGHT CONTROL: this is very interesting in advertising for example. Spotlights of appropriate power over the fluorescent surface leaves a luminescent trace that fades away with time. Usually advertising boards due to safety issues and better visibility are placed out of people's range. With the application of this control system users can control the whole process without the necessity of using heavy machinery to reach standard cable control systems. In addition, with the help of an internet connection, control over a network of advertising boards can be remotely controlled from one place.

CONNECTION BETWEEN A SENSOR AND A SERVOMOTOR CONTROL: everywhere, motion is needed as a response to the environmental data collected by the sensors (light, movement, sound, acceleration, humidity, etc.) that is out of range for servicing personnel. Another advantage is the size area that can be covered by this control system that is limited only by the range of a wireless network. In addition, the amount of control system in one infrastructure is also nearly limitless.

EDUCATIONAL, LABORATORY EXERCISE: the advantage of this control system is the simplicity of the basics. When trying to pass on to students a little bit of new knowledge, it is important not to discourage them with very difficult tasks. Here we have simple basics that provide instant visible result, which motivates new students to the subject, as well as capabilities of creating really complicated and advanced projects that demand knowledge and outside the box thinking from students familiar with the subject.

SMART HOUSES: to connect automation of the smart houses to the computer network, inside intermediary systems are needed. This is where the application of a wireless servo control systems comes in. The next step in smart housing is to be able to control blinds, heating, security, sensors, etc. not only from on-wall panels inside the house but also from a computer or mobile device through a computer network or even remotely via internet.

SECURITY: another place to adopt a designed control system is security. This is where many connections between motion and data from various sensors are found. As we all know the main issue in this application is safety. In this case highly encrypted wireless connections between a control system and access points as well as additional encryption algorithms inside of the control system are needed. All of the requirements for this application are fulfilled and new security algorithms can be adopted.

INDUSTRY CONTROL OVER AUTOMATION OF PRODUCTION PROCESS: this is the application where a wide area must be covered by many independent control systems responsible for separate processes. In this case adopting a wireless control limits the amount of work for the personnel that directly leads to the reduction of the staff and increases monthly savings.

Implementation

Implementation of the conception presented above was divided into two separate parts. The first part contained

hardware, and a main box (Arduino Uno + WiFi shield) with WCS onboard, and the preparation of two servomotors and creating a BlackBox with photocell and servo connected to the blind inside.

The process of creating the hardware aspect of the project was not complicated. In the first part (the main control box) both electronic boards were nearly completed, the only two things left to do were soldering the headers to the WiFi shield and creating a reset button with a corresponding LED, connecting pins for servos and photocells and casing for the antenna. Until the project was finished all ideas were prepared and tested on the breadboard.

All the casings were made from 3mm plexi plates. Main casing – transparent, BlackBox – black.

A reset button was added as a safety measure that returns all network settings to default to avoid having to reprogram the mainboard in case the wrong settings were set.

Second was the software which was also divided into two sections, control board and PC. A block diagram of control system software is presented in figure 2.

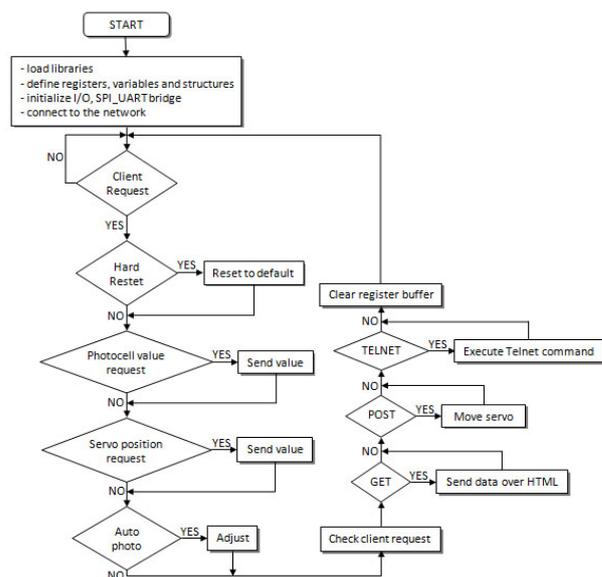


Fig. 2. Block diagram of control system software

After start up and setting up all needed libraries WCS can be set into two modes. One, when awaiting for a specified order and sending back to the PC values measured by the sensors and second, where automatic mode is on. In this case WCS maintains the light intensity inside the BlackBox on a defined level. This mode shows that this control system is not only prepared to set value we send to it but also can supervise the whole operation returning state of the operation to the PC application connected.

For programming, a mainboard environment called Arduino alpha is provided. It is an open project written, debugged and supported by Massimo Banzi, David Vaurtilles, Tom Igoe, Ginaluca Martino and David Caurtielles. It is based on "Processing" by Casey Reas and Ben Fry. As the entire language is mostly based on C++ this process was limited to recognize syntax distinctive for Arduino board. The "Learning" section of the Arduino project website contained essential code examples and appeared to be very useful in understanding how all of this works.

After a few days of playing with Arduino, it was time for the WiFi shield. Everything went smoother as in this case more technical support was provided.

The last step was application of the BlackBox. To do this the handling of analog input as well as two modes of work,

manual (manual control over servos) and automatic (setting light intensity without controlling over the servo) were needed.

The entire syntax of Arduino code consists of 4 parts:

- I. Declarations of libraries, variables, structures and registers that are needed. For example:
 - External libraries:
 - o string.h – responsible for string commands
 - o servo.h – set of commands controlling servo-mechanisms
 - o EEPROM.h – allows us to store data in 512 bytes of EEPROM memory where values are stored when the board is turned off
 - SC16IS750 register definitions [14]
 - SPI pin definitions
 - Global variables
 - SC16IS750 communication parameters and structure
- II. “void setup()” activated only once, at the start of the board. For example:
 - Defining servos on proper pins and resetting their position
 - SPI (Serial Peripheral Interface) to allow communication between Arduino Uno board and WiFly Shield
 - Checking in SPI-to-UART bridge is initialized successfully
- III. “void loop()” function which is called over and over again by the Arduino itself. This is the place for the main program.
- IV. Functions and procedures that we need for the board to serve its purpose from part II or III.

PC application arrives in the second stage. Delphi is the object-oriented programming language based on Pascal. The version that has been used was 2009 Professional that allows to test all means necessary for the project. Permission from the official Embarcadero distributor in Poland to use the trial version was granted for this stage.

For the application to serve its purpose a set of requirements had to be met. To describe the whole application we must start with operating principles. WCC application has two tasks: to control servo with the BlackBox and to manage WCS settings that allows it to connect to the wireless network. To achieve this we cannot forget about assigning an appropriate security level. Operating principles are very simple. At the beginning, a user needs to log in. After positive verification the application tries to connect via telnet protocol with WCS (IP address is written in an external setup.ini file). If the connection was successful, control options and settings are available.

During the writing process, the following components were used:

idTelnet – responsible for handling telnet protocol,
MainMenu – simple menu component,
Timet – controlling “on time” events in application.

GUI (Graphical User Interface) is a single standard build with components provided by a programming environment. Nothing extraordinary was adopted in this matter, just a simple menu with Login, Servo and Settings pages that are clear for the user but not necessarily eye pleasing.

To prevent unwanted people from accessing basic login protection was implemented. To level up security, a separate algorithm to encode login and password are implemented. Both of them are stored inside “setup.ini” file next to the main execute file.

When it comes to controlling both servomotors, external pins are connected to the servomotor, and a photocell is connected inside the BlackBox. In each case two modes are available: manual and auto. In servo manual mode

servomotors will move by the desired degree as told. When entering this mode on “servo_1” the photocell automatically goes into auto mode.

In photocell manual mode, users can set searched light intensity inside the box. The system will try to adjust the blinds. If the value of light intensity needed reaches its maximum value, the application will display an appropriate message.

To control data flow an “Advanced” checkbox was created. When checked, both sent and received data will appear on the screen.

The last part of the WCC are settings, that are divided into groups. Basics, where all information needed to establish connection are stored. Advanced, where more network settings like DHCP, netmask, gateway, IP, DNS, connection password, external antenna are placed.

Observations

While conducting tests during the process of projecting and creating a Control System, we have noticed a series of pros and cons. Some of them are to be revised and some are not. Knowing just a few is rather significant.

LAG: as expected in every wireless system visible lag occurs. In our case the time from sending a command from WCC software and receiving back confirmation of the completed task is about 0.8 seconds. When we take into consideration that this data flows over a WPA2-PSK encrypted network, the lag is not that long. To decrease this delay, optimization of the Arduino code is needed.

MAX SERVOS: for the demonstration of the control system we have implemented control over two servomotors. In practice the only obstacle that limits number of servos available are pins and power supply. In this case 10-13 pins are used by the WiFly shield to communicate with Arduino over SPI. The second pin is used by the button and the seventh by the LED. This leaves us with a total maximum amount of 6 servomechanisms. To increase this amount, an additional shield is required.

STATUS INDICATOR: a very useful solution adopted in the WiFly shield is a set of status indicators. To control status by taking a look over a control system, three onboard LEDs are used.

PIO4, 5 and 6 are highly active and can be connected to external LEDs to provide network, connection and data status.

RANGE: in a tested prototype system wireless 802.11g network is used. In accordance with the wireless router specifications, the device does not exceed a range of 100m indoors or 400m outdoors. As always those values are perfect. In reality, the range did not exceeded 100 meters outdoor in building area.

PHOTOCELL ACCURACY (daylight, bulb, fluorescent lamp): during testing we have noticed that the type of light source from which we are trying to measure light intensity has great influence on the results. When we use a bulb the light flickers with such an intensity that despite constant distance and light intensity the reading from the photocell in extreme cases was dropping and rising in range of ± 15 points. In the case of a fluorescent lamp the light intensity pulses causing a reading of ± 5 points when daylight reading was nearly constant. To overcome this effect, but not at a huge cost of accuracy, the delay at a level of ± 5 points to servo reaction has been added.

SPEED CONTROL: as this project is based on open source platform many beta or open libraries are available. One of those is VarSpeedServo library which allows us to easily control the speed of the connected servos. Due to the problems with application that appeared when the project was created, we have decided not to use this library. Con-

sidering rapid development of libraries it will be a very valuable asset in future modification of WCS.

ELECTRIC DISCHARGE: another advantage that is really important in application of a control system is safety from electrical discharges. When power is supplied from the source and is separate from the power network, there is no risk in electrical discharges from the grid. This allows us to omit expensive devices which are designed to protect from power network influence.

RESET: two reset buttons are placed on the control system. One (silver/gold) acts like turning off and on again. It is meant to be used when WCS hangs up. The other one (black) is designed to reset to the default settings of the wireless module. Both of them appear to be working without any problems.

ANTENNA: to expand the range of the device, an external antenna is provided. In the prototype a standard 2dBi D-Link antenna is used. As standard connection is used there is a possibility to change the antenna to one with greater range.

HANGING UP: unfortunately a prototype device is not flawless. During test specific situations, WCS tended to hang up and needed to be reset. The main issue is insufficient power supply. When a lack of power appears even for a brief moment, communication between the computer software and control system becomes impossible.

FUTURE IMPLEMENTATION: as we can see in the overview of the WiFly shield there are many settings in our Control system. The most valuable appears to be power management. When using a portable edition of this sort of device, power consumption becomes an issue. Thanks to sleep mode, wake on UART and other available functions, we can significantly extend work time on a battery. Another not fully implemented feature is the HTTP server. As an example an editor is stored inside of the Control System. It gives us opportunity to create a separate interface to manage certain settings or just show status data. Real time clock within WiFly allows us to use time stamping. Thanks to this login/session algorithms can be implemented to increase the security level of data sent over HTTP.

To allow up-to-date with firmware, there is a possibility to deliver over-the-air updates with the use of the ftp. When thinking about final product this is a worthwhile solution.

Of course there are plenty of settings not implemented in WCC software. We can connect to the Adhoc networks, manage tie or UART parameters of use UDP protocol instead of TCP/IP. With all of these possibilities highly advanced projects are at our fingertips.

Conclusion

In summary this solution implemented to resolve the problem of the wireless control system of the servomotor has met the expectations of the project. Furthermore potential future expansions leaves the solution open for development in many different directions. For example in more complicated situations, to improve the speed of operations, a golden section search algorithm can be implemented. Various new sensors like accelerometer, pressure, temperature, tilt, move or hallotron can be connected. Many communication protocols can be implemented (xbee, gsm, gps, etc.). Even bigger projects that would need up to 48 separately controlled servomotors for example are quite possible.

REFERENCES

- [1] Gieras J.F., Gieras I.A., Electrical Energy Utilization, Publishing House Adam Marszałek, 1998.
- [2] Glowacz A., Glowacz A., Korohoda P., Recognition of Color Thermograms of Synchronous Motor with the Application of Image Cross-Section and Linear Perceptron Classifier, *Przeegląd Elektrotechniczny*, Vol. 88 (2012), No. 10a, pp. 87-89.
- [3] Glowacz A., Diagnostics of induction motor based on analysis of acoustic signals with the application of eigenvector method and K-Nearest Neighbor classifier, *Archives of Metallurgy and Materials*, Vol. 57 (2012), No. 2, pp. 403-407.
- [4] Glowacz A., Glowacz W., Diagnostics of Direct Current motor with application of acoustic signals, reflection coefficients and K-Nearest Neighbor classifier, *Przeegląd Elektrotechniczny*, Vol. 88 (2012), No. 5a, pp. 231-233.
- [5] Glowacz A., Glowacz W., Diagnostics of synchronous motor based on analysis of acoustic signals with application of LPCC and Nearest Mean classifier with cosine distance, *Archives of Metallurgy and Materials*, Vol. 55 (2010), No. 2, pp. 563-569.
- [6] Glowacz A., Glowacz W., Diagnostics of dc machine based on sound recognition with application of FFT and Jacquard distance, *Przeegląd Elektrotechniczny*, Vol. 86 (2010), No. 4, pp. 292-295.
- [7] Gwozdziejewicz M., Zawilak J., Influence of the rotor construction on the single-phase line start permanent magnet synchronous motor performances, *Przeegląd Elektrotechniczny*, Vol. 87 (2011), No. 11, pp. 135-138.
- [8] Sulowicz M., Borkowski D., Wegiel T., Weinreb K., Specialized diagnostic system for induction motors, *Przeegląd Elektrotechniczny*, Vol. 86 (2010), No. 4, pp. 285-201.
- [9] Zawilak T., Investigation of higher harmonics in a Line-Start Permanent Magnet Synchronous Motor, *Przeegląd Elektrotechniczny*, Vol. 84 (2008), No. 12, pp. 122-125.
- [10] 802.11 network standards, <http://standards.ieee.org/findstds/interps>.
- [11] IEEE 802.11 Executive Committee, <http://www.ieee802.org>.
- [12] Arduino official website, <http://arduino.cc>.
- [13] WiFly User Manual, <http://www.sparkfun.com/datasheets/Wireless/WiFi/WiFlyGSX-um2.pdf>.
- [14] SC16IS750 Datasheet, http://www.sparkfun.com/datasheets/Components/SMD/SC16IS740_750_760.pdf.
- [15] Arizaga J.A., de la Calleja J., Hernandez R., Benitez A., Automatic control for laboratory sterilization process based on arduino hardware. *Electrical Communications and Computers (CONIELECOMP)*, 22nd International Conference, 27-29 Feb. 2012, 130-133.
- [16] Georgitzikis V., Akribopoulos O., Chatzigiannakis I., Controlling Physical Objects via the Internet using Arduino Platform over 802.15.4 Networks. *Latin America Transactions, IEEE (Revista IEEE America Latina)*, Vol.10 (2012), Iss. 3.
- [17] Sarik J., Kymissis I., Lab kits using the Arduino prototyping platform. *Frontiers in Education Conference (FIE)*, 2010, IEEE, T3C-1-T3C-5.

Authors:

dr hab. inż. Zygfryd Glowacz, Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej, Katedra Energoelektroniki i Automatyki Systemów Przetwarzania Energii, al. Mickiewicza 30, 30-059 Kraków, E-mail: głowacz@agh.edu.pl;

mgr inż. Jakub Wójcik, Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej, al. Mickiewicza 30, 30-059 Kraków, E-mail: jawojcik@agh.edu.pl.