
The Eurasia Proceedings of Educational & Social Sciences (EPSS), 2015

Volume 3, Pages 6-14

ICRES 2015: International Conference on Research in Education and Science

TELEOPERATION OF AN EDUCATIONAL MOBILE MINI-ROBOT VIA WIRELESS COMMUNICATION

Ognyan B. MANOLOV

Applied Informatics and Computer Technologies Dept.,
European Polytechnic University, Pernik, Bulgaria

ABSTRACT: This work aims to describe another contemporary manner for interaction between human and mechatronic device (educational mobile robot) by Bluetooth communication with a purpose for implementing wireless remote motion control. Herein the three major topics will be discussed - the educational mobile robot "Audrino"; the consoles for wireless motion control; the communication between robot and consoles.

Key words: Teleoperation, wireless communication, arduino educational mobile robot, bluetooth

INTRODUCTION

Teleoperation indicates operation of a machine at a distance. It is similar in meaning to the phrase "remote control" but is usually encountered in research, academic and technical environments. It is most commonly associated with robotics and mobile robots but can be applied to a whole range of devices or machines operated by a person from a distance, [1]. In recent years, there has been a growing interest in mobile robot motion control.

Telerobotics is the area of robotics concerned with the control of semi-autonomous robots from a distance, mainly using interfaces for Wireless network like Wi-Fi, Bluetooth, the Deep Space Network, and similar connections. Mainly to control the movement of a mobile robot we control the speed and direction of rotation of its engines by using one of the above mentioned interfaces to connect with a corresponding control circuits.

While controlling the mobile robot it has to consider that our application will not only control the robot movement but also will do additional tasks for gathering information about the environment, computing the mobile robot's moving direction and therefore the interface connection should not block these tasks. Herein the three major topics will be discussed:

1. The educational mobile robot "Audrino".
2. The consoles for wireless motion control.
3. The communication between robot and consoles.

1. CONCEPTUAL CONFIGURATION OF A DISTANCE CONTROL

1.1. The educational mobile platform "Audrino"

The Arduino robot is a self-contained platform, based on the popular BOE Shield-Bot, which includes:

- Board of Education Shield PCB
- High-quality aluminum robot chassis, continuous rotation servos and wheels
- Components needed for tactile, light, and infrared distance navigation systems..

Arduino robot is a result of the collective international team effort in collaboration with Comclubot - 4-time world champion in robocup junior robotics soccer - who looking at how science can be made fun to be learning and to be developed an interactive machine that allows exploring an environment. [2].

The Arduino hardware and software have many functional similarities to the BASIC Stamp microcontroller and software. Parallax Inc. teamed up with SimplyTronics to design the Board of Education® Shield, which makes the Arduino hardware compatible with the Boe-Bot chassis.

Both the functional similarity and the hardware compatibility made very simple porting of example programs and educational material from Robotics with the Boe-Bot to robotics with the Board of Education Shield.

- This is an Open Access article distributed under the terms of the Creative Commons Attribution-Noncommercial 4.0 Unported License, permitting all non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

- Selection and peer-review under responsibility of the Organizing Committee of the conference

*Corresponding author: Ognyan B. MANOLOV- icemstoffice@gmail.com

For our research and experiments was assembled an own “Arduino” mobile robot which is shown on Fig. 1. It is based on classical two wheeled platform with reversible DC-servo driven wheels and spherical fulcrum on the rear side. The Motor control module drives the motors, and the Control Board reads sensors and decides how to operate. Each of the boards is a full programmable using the Arduino IDE.

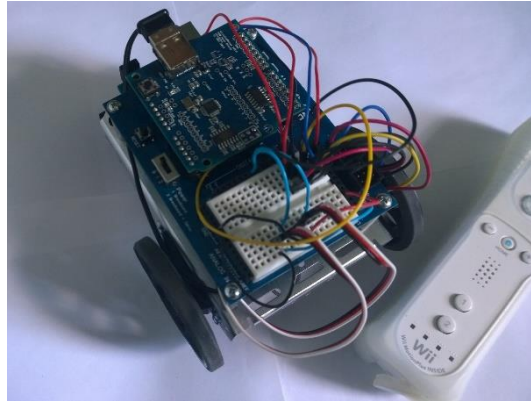


Fig. 1. The Assembled Own “Arduino” Type Mobile Robot

The robot’s completeness includes the following modules: *Arduino Uno Rev3*, *Parallax Robotics Shield Kit*, *USB Host Shield 2.0* and *Bluetooth* communication, which modules are described in detail as follows below.

1.1.1. The Arduino Uno Rev3 module. This is an open source microcontroller board, represented on Fig. 2, based on the Atmel ATmega328 MCU, plus a free software development environment [2]. The module can be used to sense inputs from switches, sensors, and computers, and then to control motors, lights, and other physical outputs.



Fig. 2. The module Arduino Uno Rev3 (front and back)

It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The *Arduino Uno Rev3* differs from other preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter. The microcontroller board can be powered via the USB connection or with an external power supply. The power source is selected automatically. The technical specifications of the module *Arduino Uno Rev3* are:

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB (ATmega328) of which 0.5 KB used by boot-loader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz

1.1.2. The Parallax Robotics Shield Kit module. Parallax Inc. introduced the original Boe-Bot® robot with its BASIC Stamp® 2 brain shown below in 1999, [3]. The module includes a Board of Education Shield (BOE), represented on Fig. 4, which makes it easy to build circuits and connect servo motors to the *Arduino Uno Rev3* module. The BOE Shield mounts on metal chassis with motors and wheels.

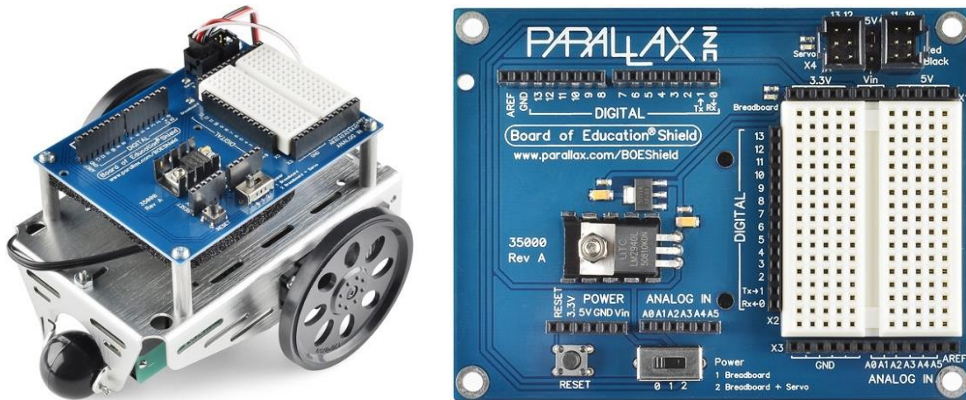


Fig. 4. The Parallax Robotics Shield Kit and BOE

With this kit and Arduino module is able to activate over 40 hands-on activities in robotics, such as:

- Learning to program the robot's Arduino Brain
- Calibrating the robot's continuous rotation servo motors
- Using lights and speakers for status indicators
- Assembling the robot
- Preprogrammed navigation
- Using touch-switches to navigate by contact with objects
- Using phototransistors to navigate by light
- Using non-contact infrared sensors to measure distance and avoid or follow objects
- Remote motion control by wireless communication.

1.1.3. Arduino Software. The Arduino integrated development environment (IDE) is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the wiring projects. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a "sketch".

The Arduino free programming software is designed for communication with a computer, another Arduino, or other microcontrollers by UART TTL (5V) serial communication and includes a serial monitor which allows simple textual data to be sent to and from the Arduino board, as it is shown on Fig. 3. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer.

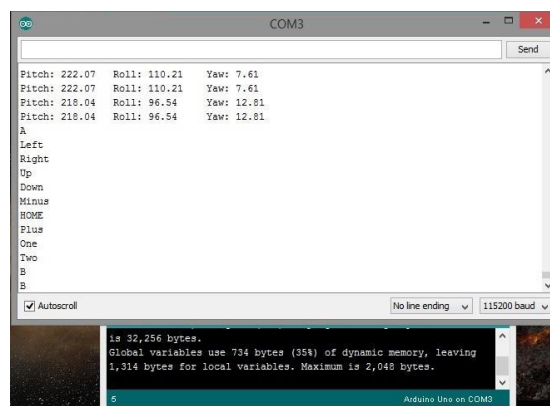


Fig. 3. A Textual Data Sent To And From The Arduino Board

The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the boot-loader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer.

Arduino programs are written in C++. The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier. Users only need define two functions to make a runnable cyclic executive program:

- **setup():** a function run once at the start of a program that can initialize settings;
- **loop():** a function called repeatedly until the board powers off.

2.2. The consoles for wireless motion control

The concept of the mobile robot's teleoperation by with "Wii Remote" console has been presented at the „Robotics and Mechatronics Festival 2014" in the Ruse University, Ruse, Bulgaria [4]. The idea evolved in developing two more forms of control. The second was with Xbox controller via Wireless Gaming Receiver and third with mobile application for Windows Phone.

2.2.1. The "Wii Remote" console (handle). In 2006, Nintendo released a console for home entertainment called Wii. It fast became popular thanks to its intuitive motion controllers, Wii Remote and NunChuk. The console Wii Remote, informally known as the Wiimote, is the Wii's main input device. It was not long before enthusiasts started to use them for different applications. The Wiimote handle is a wireless device, using standard Bluetooth technology to communicate with the Wii. This handle is able to communicate wirelessly with the controller via short-range Bluetooth radio, which permits to operate up to 10 meters away from the console with up to four controllers. However, to utilize the pointer functionality, it must be used within 5 meters [6].

It uses the standard Bluetooth Human Interface Device (HID) protocol to communicate with the host, which is directly based upon the USB HID standard. As such, it will appear as a standard input device to any Bluetooth host. The Wiimote actually uses a complex set of operations, transmitted through HID Output reports, and returns a number of different data packets through its Input reports, which contain the data from its peripherals. More over, the controller's symmetrical design allows it to be used in either hand and also to use two Wiimote handles in each hand, [6]. The Wiimote handle, shown on Fig. 5, represents a one-handed, remote-control based design console, instead of the traditional gamepad controllers.

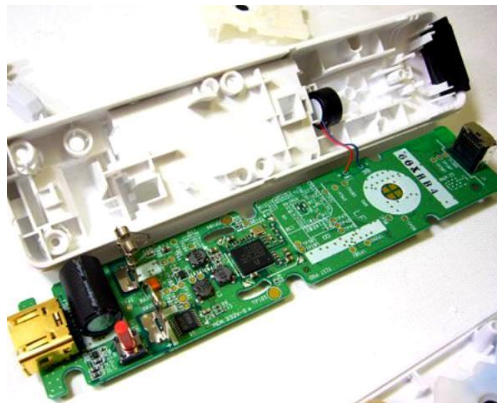


Fig. 5. "Wii Remote" One-Handed Console

The handle has the ability to sense acceleration along three axes through the use of an ADXL330 accelerometer (see Fig. 6). It also features a PixArt optical sensor, allowing it to determine where the console is pointing.



Fig. 6. The ADXL330 Accelerometer In "Wii Remote"

Wiimote has a built into the console Broadcom BCM2042 Bluetooth System-on-a-chip, which contains multiple peripherals that provide data to it, as well as an expansion port for external add-ons. The **BCM2042 microcontroller**, shown on Fig. 7, includes a large 108 Kb on-chip ROM section for storing firmware.



Fig. 7. "Wii Remote" With The BCM2042 Microcontroller

The Wiimote contains a 16 KB EEPROM chip with a section of 6 kilobytes to be freely read and written by the host. If the EEPROM chip really contains code for the BCM2042 then this was probably done to make firmware updates possible, so there might be a way of accessing the other parts of the EEPROM via *Bluetooth* as well.

2.2.2. The “Wii MotionPlus”. This is an expansion device for the Wii Remote (see Fig. 8) that allows it to capture complex motion more accurately, as a remote design is fitted perfectly for pointing, and in part to help the console appeal to a broader audience that includes non-gamers.

“Wii Remote Plus” as a Motion controllers are used to achieve some desired benefits which can include:

- increased the accuracy of position and speed;
- higher speeds;
- faster time of reaction;
- increased productivity;
- smoother movements;
- integration with other automation;
- integration with other processes;

It incorporates a *dual-axis tuning fork gyroscope*, and a *single-axis gyroscope* which can determine rotational motion [7]. The information captured by the angular rate sensor can then be used to distinguish true linear motion from the accelerometer readings.



Fig. 8. Wiimote With The Expansion Device “Wii Motionplus”

This allows for the capture of more complex movements than possible with the Wiimote alone. More over, it gives the ability the Wiimote also to be turned horizontally and used like a steering wheel.

2.2.3. X-box controller via Wireless Gaming Receiver. The adapter was first revealed at E3’2006 and released on February 16, 2007 [9]. Xbox 360 wireless accessories communicate over a proprietary 2.4 GHz protocol.

In order to accommodate this, Microsoft released the Wireless Gaming Receiver, which allows wireless Xbox 360 accessories to be used on a Windows-based PC or in this case on an Arduino, [8].

Both wireless devices, the Wireless Gaming Receiver and the Xbox 360 Controller, are represented on Fig. 9. After connecting to a mobile robot left analog stick controls steering, right trigger moving forward, left trigger moving backwards.

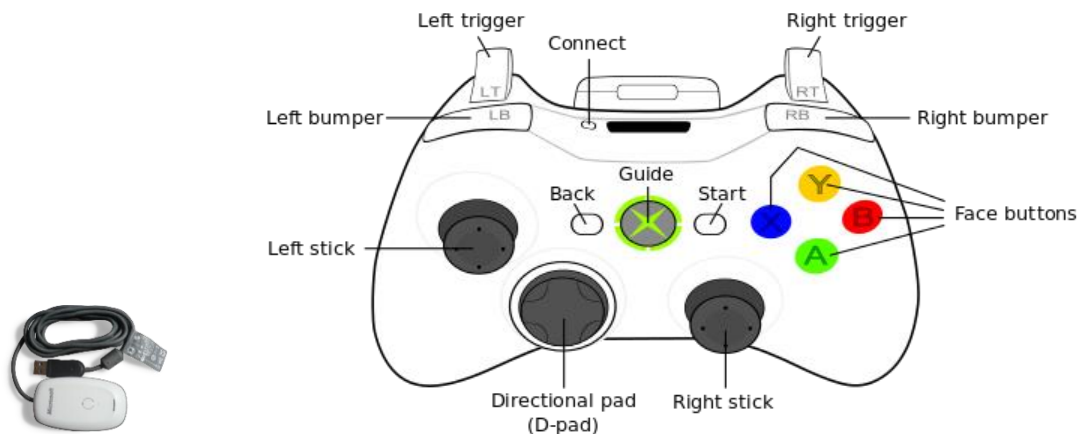


Fig. 9. The Wireless Gaming Receiver and Xbox 360 Controller Layout

2.2.4. Windows phone. Windows Phone is Microsoft's the operating system for smart phones. It is the successor to Windows Mobile, but there is no backward compatibility. Microsoft created a new flat user interface, featuring a design language named "Modern" and deep integration with social networks and all Microsoft services, first launched with Windows Phone 7.

Windows Phone 8 is a fresh start for the platform: Microsoft has abandoned the old stack of technologies used in Windows Phone 7 (the Windows Mobile kernel, Silverlight, XNA) to embrace the new features introduced in Windows 8, like the new kernel, the Windows Runtime, and the native code (C++) support [10].

The official platform to develop Windows Phone applications is Visual Studio 2012, although support has also been added to the Visual Studio 2013 commercial versions. The major difference is that while Visual Studio 2012 still allows you to open and create Windows Phone 7 projects, Visual Studio 2013 can only be used to develop Windows Phone 8 applications.

To start application development it was necessary to download the Windows Phone 8 SDK. Because the Visual Studio 2012 was not installed the first step was to install the SDK and the emulator and add them to the existing Visual Studio installation [11].

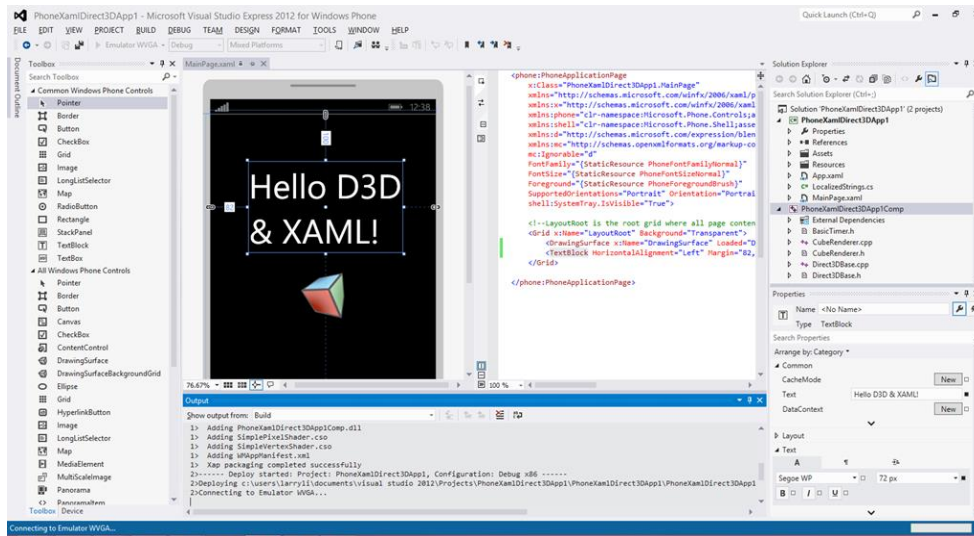


Fig. 10. The New tools for Windows Phone 8.0 SDK

The developed mobile application for Arduino mini-robot control communicates with the robot via Bluetooth. The four directional dpad for sending commands to the robot is shown on Fig. 11.

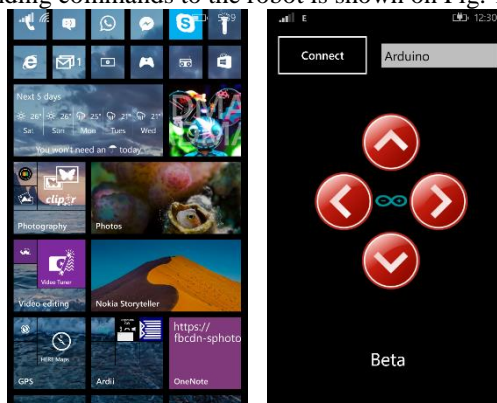


Fig. 11. Customized Start Screen of Windows Phone 8.1 and Mobile Application for sending commands

2.3. The Bluetooth communication between robot and console

2.3.1. Bluetooth. Bluetooth is a packet-based protocol with a master-slave structure, which is based on a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz) from several fixed and mobile devices and building personal area networks (PANs) to connect devices, overcoming problems of synchronization.

One master may communicate with up to seven slaves in a piconet; all devices share the master's clock. Packet exchange is based on the basic clock, defined by the master, which ticks at 312.5 μs intervals. Two clock ticks make up a slot of 625 μs; two slots make up a slot pair of 1250 μs.

A master Bluetooth device can communicate with a maximum of seven devices in a piconet (an ad-hoc computer network using Bluetooth technology), though not all devices reach this maximum. The devices can switch roles, by agreement, and the slave can become the master.

3.2.2. The USB Host Shield 2.0 module. This is a universal connection tool, shown on Fig. 12, and together with a Bluetooth dongle is used as add-on board in the Arduino developed platform in order to connect the educational mobile robot with a wireless controller and in this way to equip the mobile robot with a wireless communication. There is interference between USB Host Shield and the Board of Education Shield, so Arduino's SPI and Host Shield are connected via wires through the breadboard.

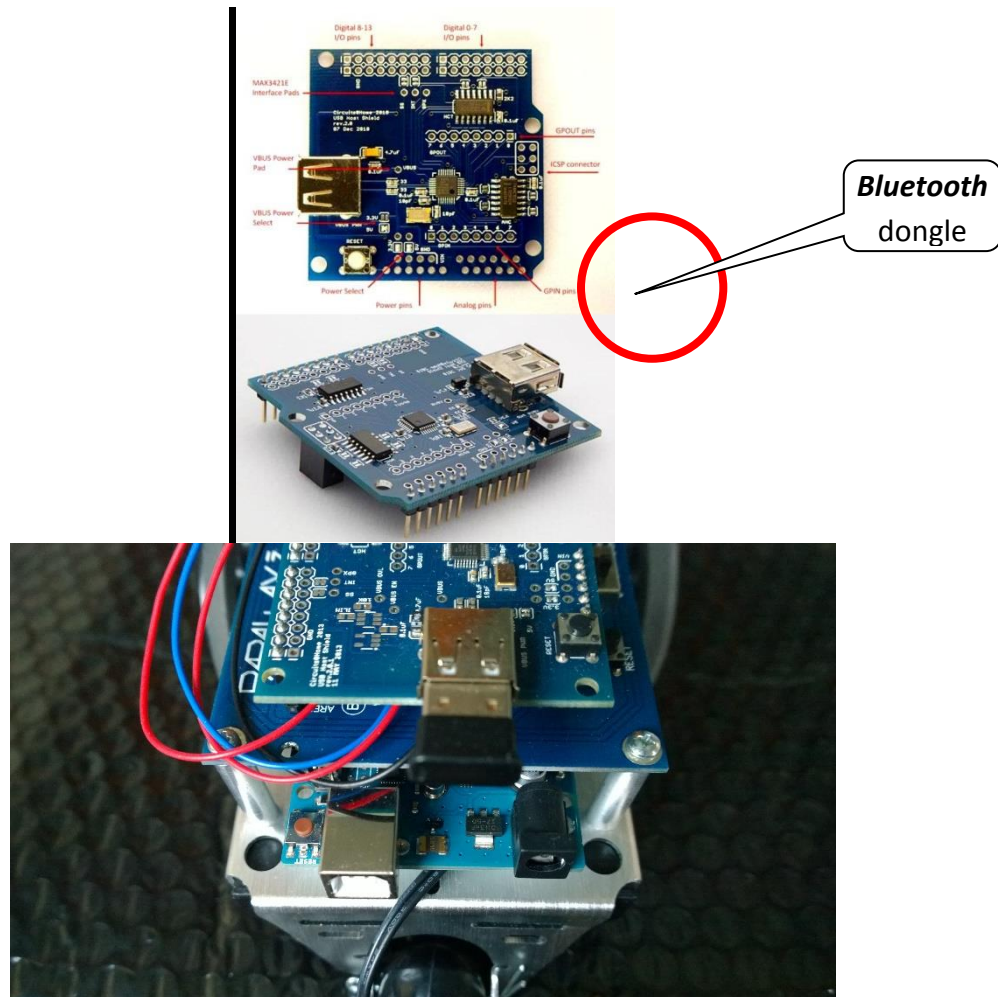


Fig. 12. The USB Host Shield 2.0 Modules And With A Bluetooth Dongle

The shield provides USB Host interface, allowing full and low-speed communication. The USB Host Shield 2.0 currently supports the following device classes [12]:

- HID devices, such as keyboards, mice, joysticks, etc.
- game controllers - Sony PS3, Nintendo Wii, Xbox360
- USB to serial converters - FTDI, PL-2303, ACM, as well as certain cell phones and GPS receivers
- ADK-capable Android phones and tables
- Digital cameras - Canon EOS, Powershot, Nikon DSLRs and P&S, as well as generic PTP
- Mass storage devices, such as USB sticks, memory card readers, external hard drives
- Bluetooth dongles.

The board contains Maxim MAX3421E USB host controller, 12MHz crystal, level shifters, resistors, capacitors, Reset button and USB A-type connector. There are also a number of solder pads and jumpers, which are marked with red arrows. MAX3421E interface pads are used to make shield modifications easier. Pads for SS and INT signals are routed to Arduino pins 10 and 9 via solder jumpers. In case pin is taken by other shield a re-routing is necessary, a trace is cut and corresponding pad is connected with another suitable Arduino I/O pin with a wire. To undo the operation, a wire is removed and jumper is closed. GPX pin is not used and is available on a separate pad to facilitate further expansion. It can be used as a second interrupt pin of MAX3421E.

The board layout is completed with : Power Select, Power pins, Analog pins, GPIN pins, ICSP connector, GPOUT pins, Digital I/O pins 0-7, digital I/O pins 8-13, MAX3421E interface pad, VBUS power pad
 For activation of **USB Host Shield 2.0** module with **Bluetooth** has to be written communication software and uploaded part of which it is shown on Fig. 13.

```

}      /* fill in setup packet */
      setup_pkt.ReqType_u.bmRequestType = bmReqType;
      setup_pkt.bRequest = bRequest;
      setup_pkt.wVal_u.wValueLo = wValLo;
      setup_pkt.wVal_u.wValueHi = wValHi;
      setup_pkt.wIndex = wInd;
      setup_pkt.wLength = nbytes; ....{
    
```

Fig. 13. Part Of Communication Software For Activation Of USB Host Shield 2.0

For activation of Wiimote one-handed console with the expansion device “**Wii MotionPlus**” we need to provide a software initialization procedure where after its fulfillment the Wiimote reached to the return states, represented on Table 1.

Table 1. Software Initialization Of “Wii Remote” (Return States)

For LEDs	For buttons
0x00, // OFF	0x00008, // UP
0x10, // LED1	0x00002, // RIGHT
0x20, // LED2	0x00004, // DOWN
0x40, // LED3	0x00001, // LEFT
0x80, // LED4	0, // Skip
0x90, // LED5	0x00010, // PLUS
0xA0, // LED6	0x00100, // TWO
0xC0, // LED7	0x00200, // ONE
0xD0, // LED8	0x01000, // MINUS
0xE0, // LED9	0x08000, // HOME
0xF0, // LED10	0x00400, // B
	0x00800, // A

For the accelerometer in the expansion device “**Wii MotionPlus**” we need to provide a software initialization procedure as follows:

```

accXwiiM = ((I2capinbuf[12] << 2) | (I2capinbuf[10] & 0x60 >> 5)) - 500;
accYwiiM = ((I2capinbuf[13] << 2) | (I2capinbuf[11] & 0x20 >> 4)) - 500;
accZwiiM = ((I2capinbuf[14] << 2) | (I2capinbuf[11] & 0x40 >> 5)) - 500;
and for the gyroscopes :
gyroYawRaw = ((I2capinbuf[15] | ((I2capinbuf[18] & 0xFC) << 6)) - gyroYawZero);
gyroRollRaw = ((I2capinbuf[16] | ((I2capinbuf[19] & 0xFC) << 6)) - gyroRollZero);
gyroPitchRaw = ((I2capinbuf[17] | ((I2capinbuf[20] & 0xFC) << 6)) - gyroPitchZero); ....
    
```

2.4. Control mechanism of robot movement

The following flow chart represents the working mechanism of the robot:

