Journal of Automation and Control, 2017, Vol. 5, No. 2, 46-49
Available online at http://pubs.sciepub.com/automation/5/2/3
©Science and Education Publishing
DOI:10.12691/automation-5-2-3

Design of Mobile Application for Controlling Robosoccer via Bluetooth

Jozef Varga*, Ladislav Vargovčík, Vladimír Baláž

Prototyping and Innovation Centre, Technical University of Kosice, Kosice, Slovakia
*Corresponding author: jozef.varga.2@tuke.sk

Abstract This article describes an implementation of mobile application for controlling of robosoccer robots, it was created through online development environment MIT app inventor 2. After starting of application Robosoccer at android device you can select between Player 1 or Player 2 robot soccer. Android device can communicate with robots via bluetooth interface. The application uses an accelerometer for control direction and movement slider for speed control. Users can download application from Google Play store with title "Robosoccer" and then play soccer with robots.

Keywords: *android device, robosoccer, android application, bluetooth*

Cite This Article: Jozef Varga, Ladislav Vargovčík, and Vladimír Baláž, "Design of Mobile Application for Controlling Robosoccer via Bluetooth." *Journal of Automation and Control*, vol. 5, no. 2 (2017): 46-49. doi: 10.12691/automation-5-2-3.

1. Introduction

In this paper is described a development of Android application for controlling of robosoccer robots via Bluetooth interface. Application is created at online development environment MIT app inventor 2. Android device is sending data to Bluetooth module into the robot and arduino pro mini controlling dc motors to robot via PWM motor driver from Bluetooth receiving data, Figure 1. After the start application users can choose from connection to Player 1 or Player 2 robot. Application then initialize screen for control of robot and turn on Bluetooth on device. [1,2,4]. In Bluetooth settings is necessary pairing android device with arduino module and then pressing START button for connecting to robosoccer player. Direction of robot is controlling via accelerometer and speed via movement slider. Maximal speed of robot is 4 meter per second [3,5,9].

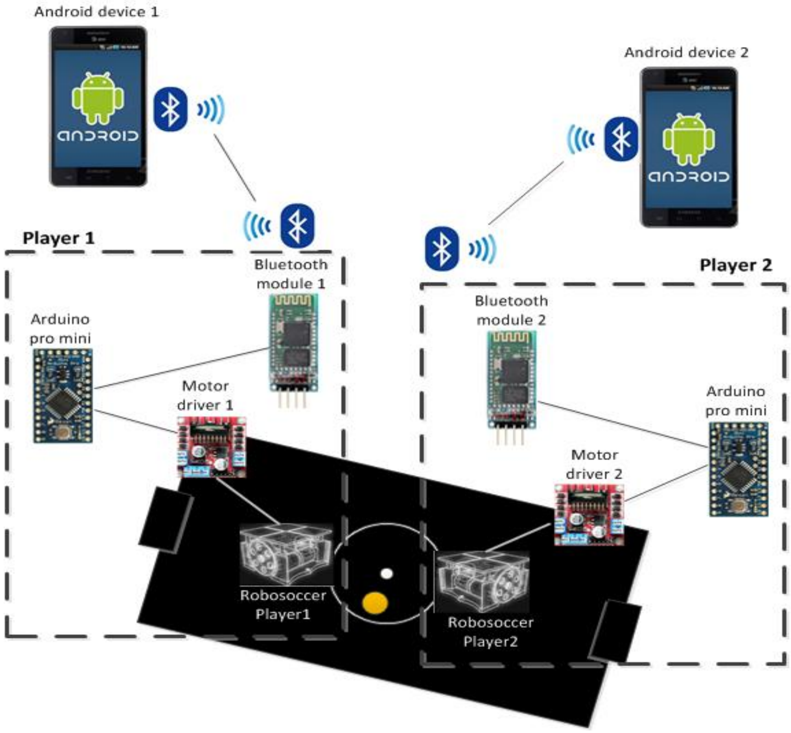


Figure 1. Interconnection of devices

2. Home Screen of Robosoccer Application

At Figure 2 is shown home screen1 of application, which allows choose from two players. The source of commands at MIT app inventor 2 was created by block schemes. At Figure 3 are shown blocks for initialization of next action. After the pressing of button Player1 will open screen2 or Player2 open screen3.

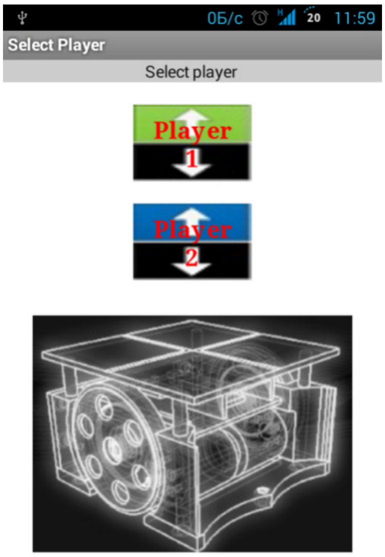
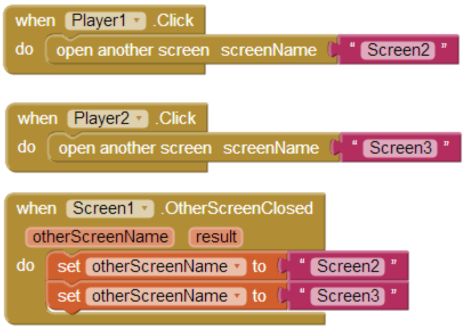


Figure 2. Home screen



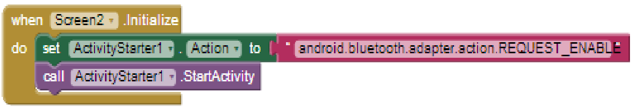
```
when Player1 .Click
do  open another screen screenName "Screen2"

when Player2 .Click
do  open another screen screenName "Screen3"

when Screen1 .OtherScreenClosed
  otherScreenName  result
do  set otherScreenName to "Screen2"
    set otherScreenName to "Screen3"
```

Figure 3. Block schemes of home screen

Bluetooth enable the activity starter for turning on Bluetooth and it is call after initialization Screen2 that is shown at Figure 4. This same activity starter will call after initialization Screen3.



```
when Screen2 .Initialize
do  set ActivityStarter1 . Action to "android.bluetooth.adapter.action.REQUEST_ENABLE
    call ActivityStarter1 .StartActivity
```

Figure 4. Call activity starter for turning at Bluetooth

3. Screens and Block Schemes for Robot Control

Screens at Figure 5 consist from Bluetooth paired Player button and from START button for Bluetooth connection to the robot. In the middle is image sprite of robot for monitoring direction and at the lower side are displayed values of accelerometer X and Y. At right side of screen is slider for speed control.

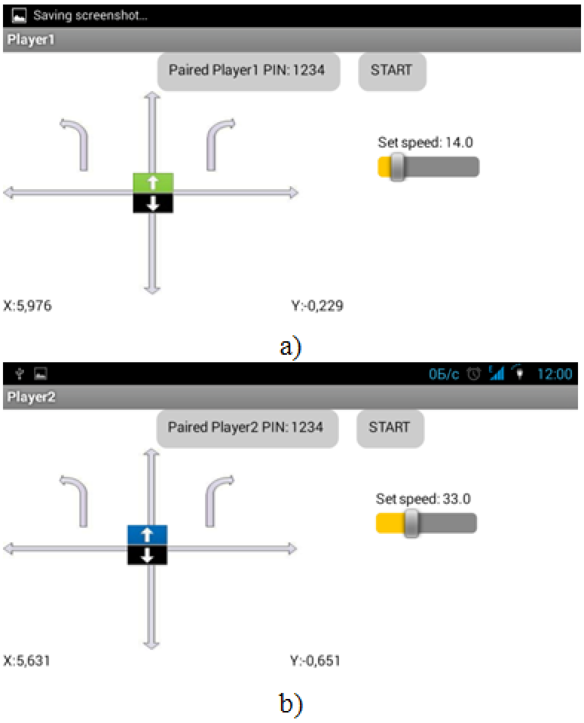
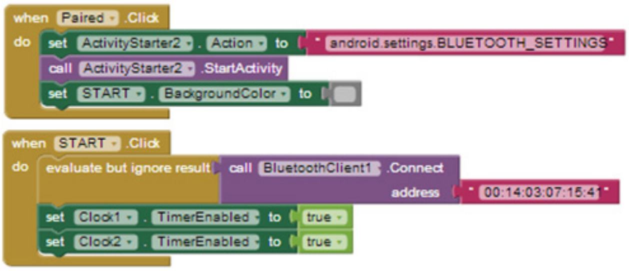


Figure 5. a) Screen2 for Player1; b) Screen3 for Player2.

By pressing of paired button we can call Bluetooth settings via activity starter that is shown on blocks at Figure 6, for pairing of android device with robot. Button START is used for connecting device to Bluetooth module which address into the robot and also for activating of Clock 1 and Clock 2 for sending data.

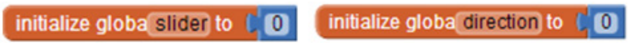


```
when Paired .Click
do  set ActivityStarter2 . Action to "android.settings.BLUETOOTH_SETTINGS"
    call ActivityStarter2 .StartActivity
    set START . BackgroundColor to

when START .Click
do  evaluate but ignore result  call BluetoothClient1 .Connect
                                                address "00:14:03:07:15:41"
    set Clock1 . TimerEnabled to true
    set Clock2 . TimerEnabled to true
```

Figure 6. Block schemes for buttons for Paired and START.

For sending data from android device to robot was creating two variables: slider and direction, shown at Figure 7. Direction is variable that is using for determination of robot direction and slider variable is using for determination of robot speed.

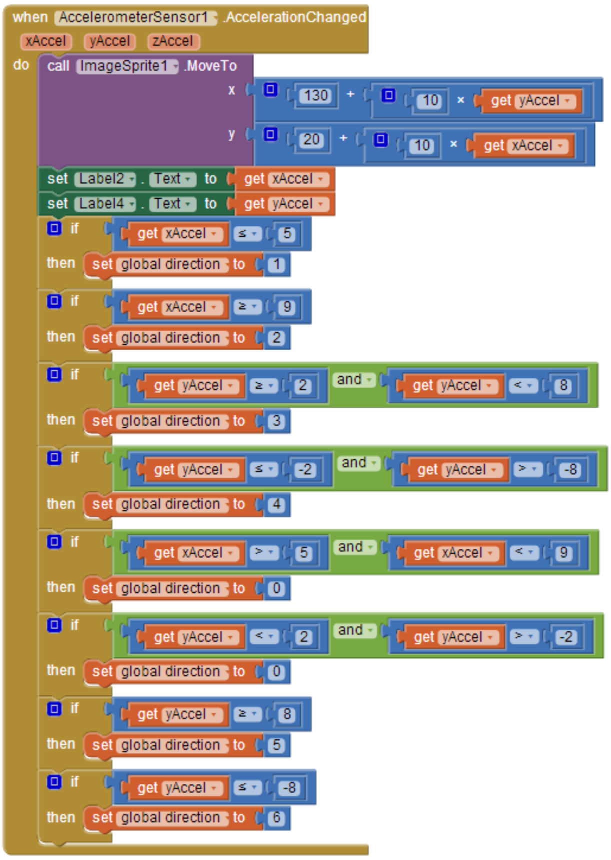


```
initialize globa slider to 0    initialize globa direction to 0
```

Figure 7. Create of slider and direction variables.

3.1. Algorithm for Sending Data of Acceleration Sensor

Algorithm that is shown at Figure 8 is using for sending data about tilting of android device and determined of zero position. On the base of this data is possible to determine direction of robot movement.

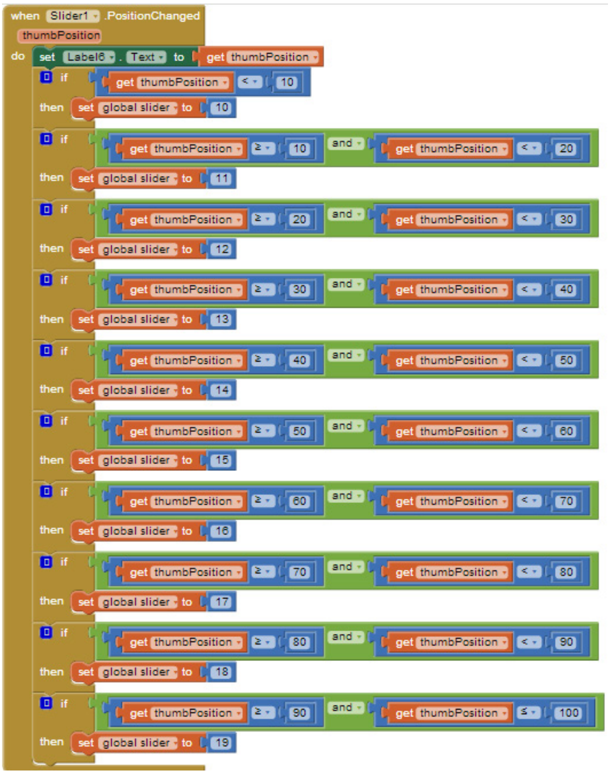


```
when AccelerometerSensor1 .AccelerationChanged
  xAccel  yAccel  zAccel
do  call ImageSprite1 .MoveTo
                x  130 + 10 × get yAccel
                y  20 + 10 × get xAccel
    set Label2 . Text to get xAccel
    set Label4 . Text to get yAccel
    if    get xAccel ≤ 5
    then  set global direction to 1
    if    get xAccel ≥ 9
    then  set global direction to 2
    if    get yAccel ≥ 2  and  get yAccel < 8
    then  set global direction to 3
    if    get yAccel ≤ -2  and  get yAccel > -8
    then  set global direction to 4
    if    get xAccel > 5  and  get xAccel < 9
    then  set global direction to 0
    if    get yAccel < 2  and  get yAccel > -2
    then  set global direction to 0
    if    get yAccel ≥ 8
    then  set global direction to 5
    if    get yAccel ≤ -8
    then  set global direction to 6
```

Figure 8. Algorithm for sending data of acceleration sensor

3.2. Algorithm for Sending Data of Slider

Algorithm shown at Figure 9 is used for slider variable assignment of thumb position. On the base of this data is set speed of robot.

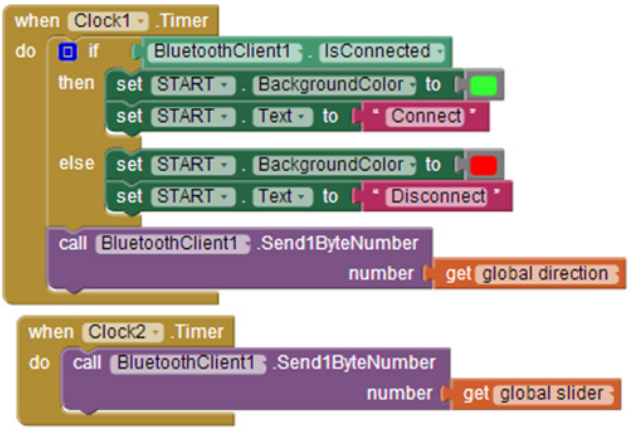


```
when Slider1 .PositionChanged
  thumbPosition
do  set Label6 . Text to get thumbPosition
    if    get thumbPosition < 10
    then  set global slider to 10
    if    get thumbPosition ≥ 10  and  get thumbPosition < 20
    then  set global slider to 11
    if    get thumbPosition ≥ 20  and  get thumbPosition < 30
    then  set global slider to 12
    if    get thumbPosition ≥ 30  and  get thumbPosition < 40
    then  set global slider to 13
    if    get thumbPosition ≥ 40  and  get thumbPosition < 50
    then  set global slider to 14
    if    get thumbPosition ≥ 50  and  get thumbPosition < 60
    then  set global slider to 15
    if    get thumbPosition ≥ 60  and  get thumbPosition < 70
    then  set global slider to 16
    if    get thumbPosition ≥ 70  and  get thumbPosition < 80
    then  set global slider to 17
    if    get thumbPosition ≥ 80  and  get thumbPosition < 90
    then  set global slider to 18
    if    get thumbPosition ≥ 90  and  get thumbPosition ≤ 100
    then  set global slider to 19
```

Figure 9. Algorithm for sending data of slider

3.3. Sending Data of Slider and Direction Variables

Clock1 is sending data from direction variable and Clock2 is sending data from slider variable, block scheme is shown at Figure 10.

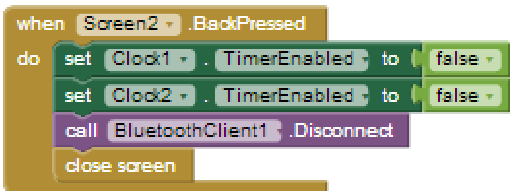


```
when Clock1 .Timer
do  if    BluetoothClient1 . IsConnected
    then  set START . BackgroundColor to 🟩
          set START . Text to " Connect "
    else  set START . BackgroundColor to 🟥
          set START . Text to " Disconnect "
    call BluetoothClient1 .Send1ByteNumber
                   number  get global direction

when Clock2 .Timer
do  call BluetoothClient1 .Send1ByteNumber
                   number  get global slider
```

Figure 10. Sending data of slider and direction variables

For closing of application is used back button of device. After the pressing of back button was call the stop for sending data from Clock1 and Clock2 and then follows closing of application, shown at Figure 11.



```
when Screen2 .BackPressed
do  set Clock1 . TimerEnabled to false
    set Clock2 . TimerEnabled to false
    call BluetoothClient1 .Disconnect
    close screen
```

Figure 11. Block schemes for back button pressed.

4. Conclusion

This application describes an implementation of android application that was created at MIT app inventor 2 for controlling of robosoccer players, made by Department of robotics SjF TUKE. Connection between android device and arduino at robot is realized by Bluetooth interface. Users can download application from Google play store – Robosoccer. After starting of application users can choose from controlling green picture – Player 1 or blue picture – Player 2 robot. Robosoccer can play two users between themselves with android device. This application can be extended with more robots and it can be used for controlling with other types of robots.

This contribution is the result of the project implementation: Výskum syntetických a biologicky inšpirovaných lokomócií mechatronických sústav v členitom teréne. Project VEGA 1/0872/16 financed by the Slovak Ministry of Education.

References

[1] Havlík, Štefan - Hricko, Jaroslav. The analysis and design study of high speed robotic devices. In Applied mechanics and materials, 2013, vol. 282, p. 18-26.

[2] Zubrzycki J., Świć A., Taranenko V.: Mathematical model of the hole drilling process and typical automated process for designing hole drilling operations. Applied Mechanics and Materials Vol. 282 (2013) pp. 221-229.

[3] Piotrowski A.: The Spectacle of Architectural Discourses," at Architectural Theory Review, (13:2, Routledge, 2008): 130-144.

[4] T. Mikolajczyk, J. Musial, L. Romanowski, A. Domagalski, L. Kamieniecki, M. Murawski, Multipurpose Mobile Robot, Applied Mechanics and Materials, vol. 282, (2013), 152-157.

[5] Slamka, J., Jedinák, M., Tolnay, M., Bachratý, M., Staš, O.: Automatic manipulation of parts made from yielding material. In: Applied Mechanics and Materials Vol. 332 (2013) pp. 432-430.

[6] Mikolajczyk T., Olaru A., Krainski P.: Adaptive control system for drill machine. In: Applied Mechanics and Materials Vol. 436 (2013) pp. 445-450.

[7] Beniak, Juraj - Križan, Peter - Matúš, Miloš: Accuracy of Rapid Prototyped models with using of FDM technology. In: Applied Mechanics and Materials. - ISSN 1660-9336. - Vol. 613 : 13th International conference on Industrial, Service and Humanoid Robotics, ROBTEP 2014, Štrbské Pleso, Slovakia, 15-17 May 2014 (2014).

[8] Beniak, Juraj: Rapid prototyping and accuracy of created models. In: ERIN. - ISSN 1337-9089. - Roč. 5, č. 6 (2012), s. 2-8.

[9] Świć A., Wołos D., Zubrzycki J., Opielak M., Gola A., Taranenko V., Accuracy Control in the Machining of Low Rigidity Shafts, Applied Mechanics and Materials, 613, 2014, pp. 357-367.