Let’s Move IT – Unity as fleet manager

*Rene Bakx, April 2018*

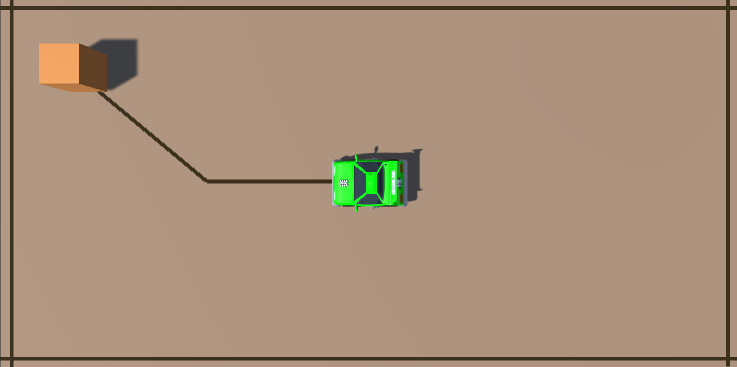


Table of contents

1. Introduction 3

2. Bibliography 4

Appendix 1 Connection of Unity to EV3 Mindstorms 5

# Introduction

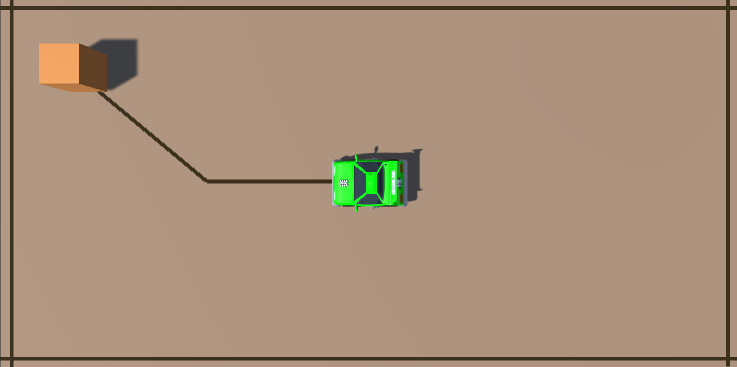
In this document the work on Let’s Move IT [1] is described. This work is part of the Fontys FHICT activities on the two work packages: WP1 (planning and control) and WP4 (test area mobile robots).

The main research question to be answered:  
“*Can Unity be used as a fleet manager for simple mobile robots*?”

The use of Unity and simple robots enables FHICT Technology students to participate in a practicle way in this research. A test area will be set up for simple robots like LEGO Mindstorms EV3. These robots will be controlled by Unity by setting up a gameplay in which the robots will have a common task to complete. This means there will be two-way real time communication between Unity and the robots. See Figure 1 below.



Figure Unity - EV3 connection



sensor data

move commands

# Bibliography

1. Kiela, H. (2017). RAAK MKB Project LET’S MOVE IT. Retrieved from <https://www.nwo.nl/onderzoek-en-resultaten/onderzoeksprojecten/i/89/29489.html>
2. Bakx, R. (n.d.). UnityRobot project. Retrieved May 17, 2016, from https://github.com/rbakx/UnityRobot
3. Bartneck, C., Soucy, M., Fleuret, K., & Sandoval, E. B. (2015). The Robot Engine - Making The Unity 3D Game Engine Work For HRI. *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN2015)*, *RO-MAN2015*, 431-437. doi:10.1109/ROMAN.2015.7333561

# Appendix 1 Connection of Unity to EV3 Mindstorms

Connecting the EV3 Mindstorms robot to a host like Unity is not trivial. Several options are investigated. A difficulty is that Unity 2017 4.0f1 supports .NET 3.5. This means that existing code or libraries using .NET > 2.0 (which most do) cannot be used within Unity.  
**At the Build conference 2016 Microsoft announced that Unity will join the .NET Foundation, so in the future it will be easier to use existing code and libraries.**

**Connecting using Bluetooth and EV3Messenger**

Making use of <https://ev3messenger.codeplex.com/> it is possible to connect a host to the EV3 using Bluetooth. However, some .NET 4 functionality is used: the queuing mechanism in System.Collections.Concurrent. So this has to be replaced. In Windows an applications connects to a Bluetooth device via a serial port emulation. On the Mac OS X MonoDevelop-Unity is used which does not support this, so it is not possible to use the EV3Messenger code with Unity on a Mac.

Advantage of using Bluetooth to connect to the EV3 is that mailbox messaging is supported for sending and receiving. Using Wifi, sendig to a EV3 mailbox is supported, receiving seems not to be supported.

**Connecting using Wifi and MonoBrick**

Making use of <http://www.monobrick.dk/> it is possible to connect a host to the EV3 using Wifi. For the EV3 only the 'NETGEAR WNA1100 - N150 Wireless USB Adapter' is supported. It is tested that with MonoBrick it is indeed possible to send mailbox messages to the EV3 using Direct Commands. The MonoBrick library uses .NET 4 functionality so will not work with Unity. Receiving messages seems not to be supported. However, with Direct Commands it is possible to retrieve the sensor data. Also it is possible to read memory. It still has to be investigated whether it is possible to write the same memory from a standard EV3 program. If this is the case, it is possible to send messages to an EV3 using a mailbox and receive back messages through memory.

**Connecting using Wifi and the LEGO MINDSTORMS EV3 API for .NET**

Instead of MonoBrick, one can make use of <https://github.com/BrianPeek/legoev3>. These libraries also make use of .NET 4 functionality so will not work with Unity. I tested this only using a USB connection which worked.

**IMPORTANT NOTE**: When building the TestApplication provided by MonoBrick it might be that when starting the error message “Error: Failed to open connection” appears. Resolution: set Platform target from ‘Any CPU’ to ‘x86’ and (if needed) copy hidapi.dll and hidapi.dylib from the LEGO Software installation to the Monobrick Test Apllication.

**Connecting using Wifi and a simple TCP/IP connection**The EV3 can connect throug Wifi. Only the 'NETGEAR WNA1100 - N150 Wireless USB Adapter' is supported. How connect a host to EV3 through Wifi is described at <http://www.monobrick.dk/guides/how-to-establish-a-wifi-connection-with-the-ev3-brick/>. In short:

* When the EV3 wifi is enabled and connection is made to the network, the EV3 starts broadcasting an UDP message every 10 seconds. This broadcast contains the serial number of the EV3.
* The host now knows the EV3 IP address and serial number and replies to this message to let the EV3 know it can expect a connection request.
* The host sends a TCP/IP connection request.
* The EV3 accepts the connection.
* The host sends an unlock message using the Serial number of the EV3.
* The host can now send commands to the EV3 using the TCP/IP connection.

The above connection sequency can easily be implemented in C# with .NET 3.5 functionality which is used in Unity.

**IMPORTANT NOTE**: When running the C# application on a VirtualBox virtual machine, be sure that the EV3 robot is on the same network as the virtual machine. This means that in VirtualBox ‘Devices -> Network -> Network Settings -> Attached’ must be on ‘Bridged Adapter’ and not on ‘NAT’.

**Sending messages from host to EV3**

After the connection is established System Commands or Direct Commands can be sent to the EV3. One of the commands is 'WRITEMAILBOX' which can be used to send a message to a receiving mailbox on the EV3.

**Sending messages from EV3 to host**

Sending back messages from the EV3 to the host through a mailbox seems only to be supported for Bluetooth, not for Wifi.

At <https://siouxnetontrack.wordpress.com/2014/08/19/sending-data-over-wifi-between-our-pc-application-and-the-ev3-part-1/> a workaround is decscribed. On the EV3 it is possible to write text to a file which can be read on the host using the opFile(READ\_VALUE) command (byte code command for the Virtual Machine) packed in a Direct Command. There also are System commands which are not handled by the VM.

Attention point with this method is that when the EV3 is writing the file, the host cannot read it. A way around this is to let the EV3 write a new value in the file after a request from the host (e.g. 'get\_distance') and then let the host close the file after which the host can read the value.

**EV3WifiLib library**

For sending and receiving messages EV3WifiLib.dll is created containing the EV3Wifi class. This library is compiled with 'Target framework' set to '.NET Framework3.5'. Next it can be included in a Unity project. In Unity, set the 'Edit -> Project Settings -> Player -> Api Compatibility Level' to '.NET 2.0'.