APPLICATION OF LOW-COST COMMERCIAL OFF-THE-SHELF (COTS) PRODUCTS IN THE DEVELOPMENT OF HUMAN-ROBOT INTERACTIONS

Ottar L. Osen
Helge T. Kristiansen
Webjørn Rekdalsbakken
Department of Information and Communication Technology
Aalesund University College
N-6025 Aalesund, Norway
E-mail: oo@hials.no

KEYWORDS

Human-robot interaction, game technology, cell phone interface, Wii Remote, iPhone.

ABSTRACT

In the effort of developing sensible ways for interaction between humans and automated equipment the use of commercial off-the-shelf (COTS) products is shown to be fruitful in the learning process. The development in the field of consumer electronics has lead to increasingly more elaborate facilities for interaction with the human user. Modern cell phones and game technology are typical representatives of this trend. In this work such equipment has been explored in the aim to achieve easy and natural interfaces between humans and

INTRODUCTION

This document presents a work on human-robot interaction and communication performed at Aalesund University College (AUC) among lecturers and students at the Bachelor of engineering program in Cybernetics. The work is integrated in the main topics of the program. A central part of this education is represented by a broad use of student projects in close cooperation with the scientific staff. The students get the means and support to investigate commercially available electronics and computer equipment related to automatic systems and autonomous devices. AUC is situated in one of Norway's most vital industrial areas with a strong cluster of ship builders, and a variety of producers of all kinds of ship equipment. This situation has resulted in a strong collaboration between AUC and many of these companies. This cooperation has given offspring to many new products for use onboard ships, examples of which are 3DOF and 6DOF motion stabilized platforms, heave-compensated winches, and remotely controlled tracking systems for light sources and cameras. Two of the companies with which AUC has the closest contact, are Rolls Royce Marine Dept. Inc. (RR) and Offshore Simulation Center AS (OSC). These two companies maintain an extensive activity in the development of ship simulators for a variety of purposes, including high-speed craft simulators, and simulators for anchor handling and crane and winch operations. OSC is today an international supplier of the most modern simulators. In this development AUC has played an important role both in mathematical modeling and in program construction (Rekdalsbakken 2006) (Rekdalsbakken 2007). An important aspect of such simulators is of course the human interface. It has to be as realistic as possible, and at the same time not too expensive. This is the background for the current work on exploring COTS products as part of the human interface towards the advanced operating equipment onboard ships (Rekdalsbakken and Osen 2009). The two main products explored in this context are the Wii Remote controller from Nintendo and the iPhone from Apple. The experiments have been performed on small-scale models of remotely controlled vehicles and include a total communication chain from the human operator to sensors and actuators installed on the vehicles. It is the intention that these experiments shall give insight into new developments in the field of human-robot interactions, which may lead to the integration of said technologies into products that are used in real user interfaces onboard ships.

INTERFACES

The Wii Remote Controller

The Wii Remote controller from Nintendo (2010) was developed to give the player a new dimension of interactivity with games. It may be said that it has revolutionized the human feeling of being part of the game context. The Wii Remote interface consists of several parts; a 3-axis accelerometer, an infrared camera, vibration function, and several buttons for input signals. It communicates over a wireless connection using the Bluetooth (Bluetooth 2010) protocol. The Wii Remote is perhaps the best example of gadget that has become a universal control tool with a wide range of applications.

Proceedings 24th European Conference on Modelling and Simulation ©ECMS Andrzej Bargiela, Sayed Azam Ali David Crowley, Eugène J.H. Kerckhoffs (Editors)
ISBN: 978-0-9564944-0-5 / ISBN: 978-0-9564944-1-2 (CD)

The iPhone

The iPhone from Apple (2010a) has set a new trend in modern cell phone development. It is a stand-alone device for telephone and general internet communication and services. It also offers a lot of auxiliary tools and facilities, including GPS and a 3-axes accelerometer.

From a HMI perspective the iPhone is very interesting since it has a high resolution color multi touch screen. Although the idea of multi touch dates back to mid 80's (Lee, Buxton and Smith 1985) multi touch technology is still in its infancy and usage patterns established amongst users of mass market devices such as iPhone will impact the future of multi touch technology. Multi touch is probably best known to the public through Sci-Fi TV series and movies such as "Minority Report", but the growing market penetration of products such as iPod, iPhone and other smart phones are rapidly making multi touch a part of everyday life for millions. Multi touch is subject to substantial research activity and IC manufacturers are inventing new multi touch devices such as Atmel's (2010) maXTouch technology.

The iPhone communicates on common standards like wireless local area networks (Wi-Fi), Bluetooth and USB. With its comprehensive set of interfacing facilities the iPhone represents a great potential as a tool for human-robot interactions.

Communication standards

Several kinds of communication standards are involved in these experiments on human-robot interactions. The handheld devices for consumer applications are equipped with wireless interfaces like Wi-Fi and Bluetooth. These facilities have been utilized in this work in addition to infrared remote control and USB connection. In the communication chain between the different parts in the experiments Wi-Fi and radio communication (RC) have been used in addition to cabled Ethernet.

Controllers

In these experiments several kinds of embedded microcontrollers have been explored. Especially three controllers have been much used, because of their qualities regarding communication, concurrent programming, and the diversity of interfacing possibilities. These controllers are the Muvium (2005) SBC65EC SBC from Modtronix, the Javelin Stamp Demo Board (JSDB) from Parallax Inc (2006) and the Arduino (2009) Duemilanove ATmega328. These controllers have been built into the test equipment, i.e. the remotely controlled vehicles and the communication links, and perform the communication tasks in addition to controlling the motion of the vehicles and their peripherals.

HUMAN – ROBOT INTERACTION

Wii Remote and Bluetooth

Bluetooth has become a very popular wireless communication standard, especially among producers of consumer electronics. Bluetooth follows the IEEE 802.15 standard for Wireless Personal Area Network (WPAN) and operates in the 2.4 GHz frequency band. Power class 2 devices (which is the most common) has a maximum sending power of 2.5 mW and has a reach of 10 meters. For the other 2 power classes 1 and 3 the ranges are 100 meters and 1 meter respectively and maximum sending powers are 100mW and 1 mW. The Bluetooth architecture uses different protocols and gives a lot of possibilities for interconnection of different kinds of electronic units. How devices use Bluetooth is standardized in so called profiles. The profiles specify dependencies, interface formats and protocol stack. There are profiles for images, printing, ISDN, fax, file transfer, hands free, headset and so on. The Wii Remote controller uses Bluetooth in its communication with other devices. The profile used is called Human Interface Device Profile (HID). HID is used for mice, joysticks, keyboards and similar devices. HID is designed for low latency. In addition the Wii Remote controller includes an infrared (IR) camera. The camera is used to decide the position of the Wii controller relative to two fixed IR beacons (known to Wii users as the sensor bar that resides on top of their TV set, but which in fact is just 2 IR diodes placed in each end of the bar). The sensor (camera) is inside the Wii Remote.

Wi-Fi Communication and the iPhone

Wi-Fi is a common term for wireless local area networks based on the IEEE 802.11 standard (IEEE 2007). To use a Wi-Fi one needs a device with a wireless network card and a wireless access point. The iPhone has implemented this network as an interface in its human-machine interconnections. This interface is used to collect the information from the iPhone on a PC for use in the control of a remote vehicle.

VEHICLE COMMUNICATION

Wi-Fi Network

For the remote communication between the central PC and the performing equipment wireless connections have been used. The Wi-Fi network for communication with the remote vehicles is implemented on the vehicle by use of a Nano WiReach (Connect One 2010) serial to Wi-Fi bridge. This very small component is compatible with the wireless network standard 802.11b/g and can roam between different networks. The WiReach is based on the chip sets iChip CO2144 IP and Marwell 88W8686. There is no need for software drivers to use this component, and it is easily configured over a serial communication from a terminal program.

Radio Communication

Another communication standard used in these experiments is the traditional 433 MHz radio connection. This is established by the use of an RF transceiver of the type ER400TRS from Easy-Radio (LPRS 2005). This is a half duplex FM (FSK) radio transceiver with 10 mW sending power and a typical range of 250 meters. The component functions as a transparent serial communication link. A 50 ohms antenna was mounted on the transceiver output port to improve signal reach.

SYSTEM INTEGRATION

Connecting the Pieces

The overall purpose of this work has been to purchase and test the COTS products necessary to build and operate remotely controlled automatic equipment. These devices are characterized by a chain of operations reaching from human interactions, through wireless communication, to automatic control and data acquisition.

Software Integration

Several programming tools and libraries have been explored in this investigation. The programming languages used have been Java and C++, with different IDEs adapted to the different controllers and situations. An aim has been to find free open source software, which could be modified and further developed to suit the specific needs of the operations.

WII REMOTE AS A HUMAN-ROBOT INTERFACE

Wii Remote as Interface

In this project a Wii Remote control was used as interface for controlling the motion and functions of a radio controlled (RC) car. The Wii Remote has been developed to become one of the most popular and versatile game interfaces, and its applications in all kinds of games are increasing continuously. Other console gaming platforms such as Sony's PS3 and Microsoft Xbox 360 have lately released their wireless remote counterparts. Recently these devices have found a segment as a more general interface towards other equipment, e. g. as an interface towards robots and other automatic machinery. National Instruments has released a LabView interface for the Wii Remote. In this application the 3-axis accelerometer of the Wii Remote is used as control input for the car. The Wii Remote communicates with a local PC through the Bluetooth interface.

Wii Remote Accelerometer

The Wii Remote uses the popular ADXL330 accelerometer from Analog Devices (2007). This accelerometer measures three axes with a range of +/- 3g. Since the device can measure the static acceleration of gravity it can be used as a tilt sensor in addition to measure motion.

Software and Communication

Some very useful software packages have been developed for the purpose of collecting the Wii Remote signals sent over the Bluetooth connection. Two of these packages have been explored; the Java package WiiuseJ (2009), and a Visual Studio program under Creative Common License written by Edgar Barranco (2009). The last program uses a .NET library for interaction with the Wii Remote and creates a GUI for reading the input signals. Both of these systems are open source programs. With the WiiuseJ package one can set up a WiimoteListener as an interface program towards the Wii Remote. The WiimoteListener is implemented as an interface in Java, and the methods to be used must be overrided by the application programmer. With basis in these program packages GUIs were created both in Java and in Visual Studio's C# for collecting the Wii Remote signals and forwarding them in an adequate format to the remote vehicle.

As mentioned earlier the Wii Remote uses the HID profile. Unfortunately it does not conform to the standard data types and HID descriptor. This calls for non standard drivers that have been developed by members of several internet user groups. Since Nintendo does not document this protocol the developers of said drivers must use reverse engineering to understand how the Wii Remote communicates (WiiBrew 2009). In 2009 Nintendo released an add-on to the Wii Remote called Wii MotionPluss which combined with the built in accelerometer increases the accuracy and enables tracking of more complex movements. The Wii MotionPlus add-on is a tuning fork gyroscope implemented as a MEMS (micro-electrical-mechanical system) produced by InvenSense (2008). The IDG-650 gyroscope (InvenSense 2009) measures angular velocity in the pitch/roll (X/Y) axis and ISZ-650 in the Yaw (Z) axis, see figure 1.

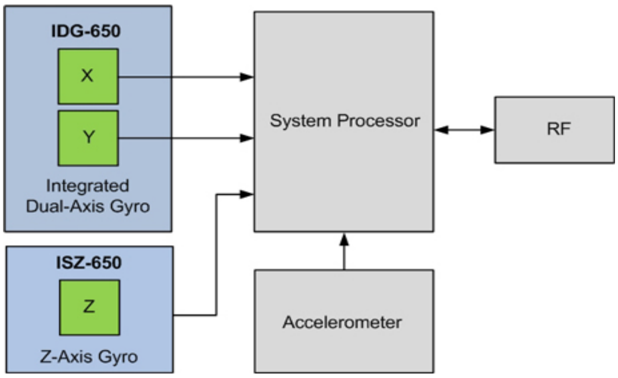

Figure 1: Game Controller System Diagram

Practical Implementations

The primary goal of this application has been to control the motion of the remotely operated car, i.e. speed and direction, by use of a handheld Wii Remote controller. In addition the application has been extended through implementation of control signals from the Wii Remote push buttons to the car. The car will be used for remote data acquisition from sensors mounted onboard the car. The communication from the PC to the remote car was implemented in two different ways, either by a local Wi-Fi network or by radio communication. The implementation of the radio link was performed by the use of two Arduino ATmega328 microcontrollers, one as a transmitter link and one as a receiver onboard the car, see Figure 2 for the transmitter device. The transmitter reads serial data from the host PC over a USB port and transfers the data to the ER400TRS radio component. This communication is fully transparent as the TX and RX pins on the Arduino microcontroller are connected both to the USB port and the ER400TRS. Thus all the data traffic goes through both these channels. It is only necessary to choose the direction of communication for the ER400TRS. In this way the data is also returned to the PC and is used as a control of the communication line. As an alternative to the Arduino microcontroller the Muvium SBC65EC microcontroller was also used in this experiment. A wired TCP/IP connection was established to the SBC65EC microcontroller. By use of the ER400TRS RC transceiver communication was established and another SBC65EC microcontroller on the vehicle. This is shown in Figure 3. Communication over the Wi-Fi network was set up directly from the PC to a Muvium SBC65EC microcontroller. This controller is furnished with a 10 Mbs Ethernet connection with a TCP/IP stack supporting socket connections for TCP and UDP. It has, however, not support for Wi-Fi communication. It was therefore decided to provide the controller with a Nano WiReach LAN to Wi-Fi Bridge. This component connects to the Ethernet port and supports wireless networks of type 802.11b/g. In this way the Muvium SBC65EC can communicate directly with a PC over a Wi-Fi network.

Program Architecture

The PC application is implemented as a program with three major classes. In Java these are realized as concurrent threads. The three classes are called *ComputerApp*, *DataSender* and *WiiMoteHandler*. The *ComputerApp* class contains the *main()* method and instantiates objects of the two other classes, which are run as independent threads. *ComputerApp* then generates a GUI for user interaction through the PC, and also establishes the communication channel to the Wi-Fi network on the given IP address and port number. A socket class connects to the given IP and port address and supports the methods for communicating on the associated input and output streams. The *WiiMoteHandler* class takes care of the communication against the Wii Remote controller over the Bluetooth transceiver on the PC. This is done by implementing the WiiMoteListener interface in the WiiuseJ library. The methods in this interface have to be overwritten. The vital method in this context is *onMotionSensingEvent()* which reads motion data from the Wii Remote's accelerometers. These motion values are verified and properly scaled before being transferred to public variables in the *ComputerApp* object. The *DataSender* class has as its responsibility to continually check for updates of these variables and transmit new incoming values to the Wi-Fi output stream. The relation between the three classes is shown in Figure 4.

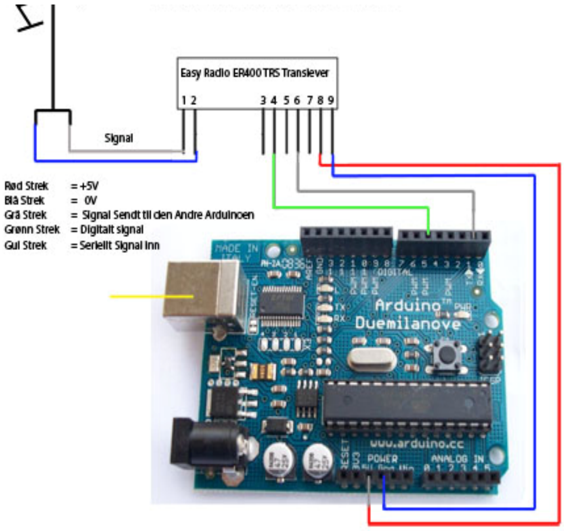

Figure 2: Arduino ATmega328 used as a RC communication link

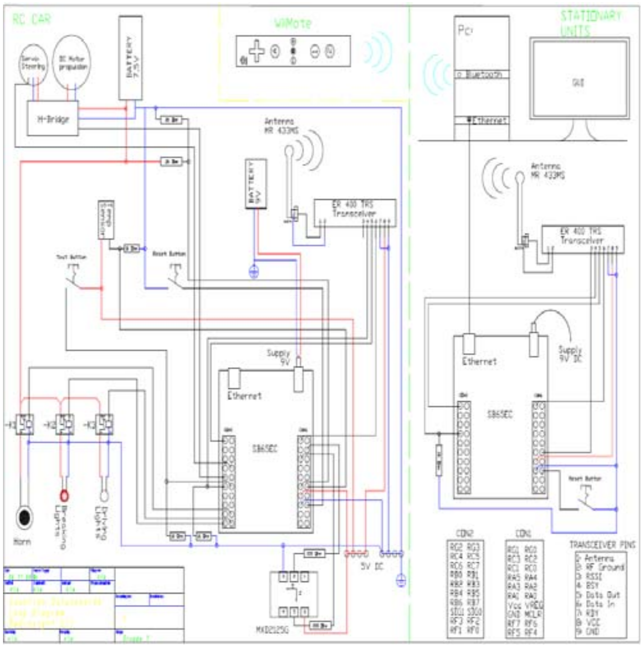

Figure 3: Electronic drawing of RC communication using SBC65EC

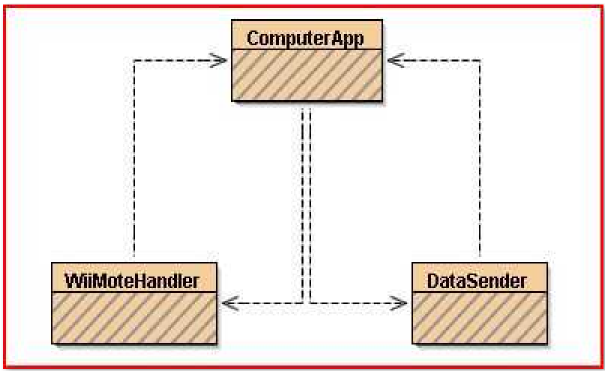

Figure 4: Class relations of the PC application

The application run on the Muvium SBC65EC microcontroller also consists of three classes; *SocketServo*, *MultiServoController* and *ServoService*. *SocketServo* runs as the main thread by inheriting from the native UVMRunnable class. First it establishes a socket connection to the Wi-Fi network on a given IP address and port, and then it instantiates objects of the two other classes. *MultiServoController* provides the methods necessary for sending PWM signals to the Muvium ports. The *ServoService* object is enabled to inherit from the native class SocketService by use of the factory method connection.addSocketService (*ServoService)*. *ServoService* implements the SerialPortListener interface and creates a Listener on the Wi-Fi connection port. This gives access to the method serialEvent(SerialPortEvent serialEvent), which starts when incoming packages arrives at the connection. These packages consist of only two bytes giving the output port number and percentage of signal value. *ServoService* forwards these values to the *MultiServoController* object for port handling.

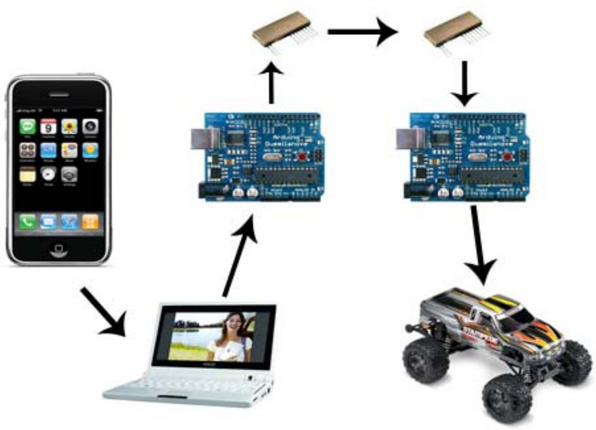

Figure 5: The communication chain.

IPHONE AS HUMAN-ROBOT INTERFACE

iPhone as Interface

In this project an iPhone was used as interface for controlling the motion and functions of a car over a wireless connection. The iPhone communicates with a PC over a local Wi-Fi network. The necessary control signals come from the 3-axes accelerometer readings on the iPhone. These data are further transmitted to the car over a wireless connection. Figure 5 shows the communication chain.

iPhone accelerometers

The iPhone uses a LIS302DL 3-axis accelerometer from STMicroelectronics (2008) with I^2C bus. This accelerometer is also able to measure static acceleration such as gravity and may therefore be used as a tilt sensor in addition to motion detection.

Software and Communication

A growing amount of public software is available for the iPhone. The iPhone has a great potential as an interface towards other equipment and this segment is still quite new and poorly investigated. Only a few Bluetooth profiles are supported (Apple 2009), hence there is a great demand for more Bluetooth support on the iPhone. As a consequence so far the focus has been on using iPhones Wi-Fi interface.

Two program systems have been used in this work; one is the OSCemote (Minor 2009) remote control application, the other is the Max5 communication package (Cycling ’74 2010). OSCemote runs on the iPhone, while Max5 is installed on the host PC. With the OSCemote program one can transmit Open Sound Control (OSC) messages over the Wi-Fi network (Freed and Schmeder 2009). OSC is a modern net based offspring of MIDI that can be used for serial communication. OSC is used for transmitting accelerometer readings from the iPhone. The Max5 program collects the data from the Wi-Fi input. It receives and unpacks UDP packages sent by OSCemote containing the accelerometer data, scales the numerical values and transmits them over a serial line to an Arduino ATmega328 microcontroller for further RC communication. Alternatively the communication goes directly over a Wi-Fi network from the PC to a Muvium SBC65EC microcontroller using a Nano WiReach LAN to Wi-Fi bridge.

The OSCemote software was selected since it is available through Apples App Store, but other software alternatives do exist and more will probably come. One interesting possibility is to write one’s own software, to facilitate that one needs the iPhone SDK which is available at Apples developer site (Apple 2010b). This approach would allow usage of industrially acknowledged protocols such as Modbus (2010) TCP.

Practical Implementations

The Max5 program collects the data from the WiFi input port. It receives and unpacks UDP packages sent by OSCemote containing the accelerometer data, scales the numerical values and transmits them over a serial line to an Arduino ATmega328 microcontroller. The ATmega328 controller operates as a transparent serial link for wireless RC communication. Another ATmega328 embedded into the remotely controlled car, receives and effectuates the radio signals. As an alternative the communication line was also realized over a WiFi network from the PC to a Muvium SBC65EC microcontroller furnished with a Nano WiReach LAN to WiFi bridge.

CONCLUSION

The primary goal of this work was to control the motion of a remote car, i.e. speed and direction, by use of commercially available handheld devices like the Wii Remote and iPhone over a wireless radio connection. However, this is only the basis for further utilizations of the car, as a tool for exploring its environment. Experiments with cameras are currently being performed and other types of sensory devices will be investigated. In this way the possibilities of using the Wii Remote and iPhone as two way communication interfaces towards remotely controlled equipment, will be investigated.

The experiments of using consumer electronics as interfaces in interactions with automatic equipment have shown that these devices have the capability to satisfy the demands raised in such situations. Many of these devices are growing more and more versatile with a comprehensive set of possibilities for easy and safe interaction with other equipment. The lack of documentation from equipment vendors may be a challenge, but that is often solved by the internet community using reverse engineering.

Due to the vast quantities these devices are produced in the prices are very low. Analysts predict sales of Wii MotionPlus at ten million units in US and Europe, and in Japan it sold something like 600 000 units the first week. The large production quantities does not only result in low cost, it also improves component quality (poor quality would have serious consequences for supplier due to the large amount of units) and finally there will be a large amount of users trying to stretch the technology to its limits and coming up with great new applications.

As a summary this work has revealed the great potential inherited in this new consumer electronic devices. The experiments encourages further work towards the implementation and use of such devices as human-robot interfaces in real ship operations.

ACKNOWLEDGMENT

We will like to express our grateful thanks to our hardworking and clever last year students in cybernetics.

REFERENCES

Analog Devices 2007. “Small, Low Power, 3-Axis ±3 g *i*MEMS Accelerometer”. Analog Devices, Inc. http://www.analog.com/static/imported-files/data_sheets/ADXL330.pdf

Apple 2009. “iPhone and iPod touch: Supported Bluetooth profiles”. Apple Inc. http://support.apple.com/kb/HT3647

Apple 2010a. “iPhone”. Apple Inc. http://www.apple.com/iphone/

Apple 2010b. “iPhone SDK 3.1.3”. iPhone Dev Center, Apple Inc. http://developer.apple.com/iphone/index.action

Arduino 2009. “Arduino open-source electronics prototyping platform”. http://www.arduino.cc

Atmel 2010. “maXTouch the worlds leading touch technology”. Atmel Corporation. http://www.atmel.com/products/touchscreens/default.asp

Bluetooth 2010. “Bluetooth Basics”. Bluetooth SIG, Inc. http://www.bluetooth.com/Bluetooth/Technology/

Barranco, E. 2009. “Wii Mote + Arduino + RC Car”. http://r00t-ed.homeip.net/projects.php

Connect One 2010. “Nano WiReach wireless LAN bridge”. Connect One Ltd. http://www.connectone.com/products.asp?did=73&pid=80

Cycling ’74 2010. “Max, Interactive visual programming environment for music, audio, and media. Cycling ’74. http://cycling74.com/products/maxmspjitter/

IEEE 2007. “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”. IEEE Computer Society, IEEE Std 802.11-2007, Revision of IEEE Std 802.11-1999.

Freed, A. and Schmeder, A. 2009. “Features and Future of Open Sound Control version 1.1 for NIME”. Conference Paper at NIME09. http://cnmat.berkeley.edu/node/7002

InvenSense 2008. “INVENSENSE IDG-600 Motion Sensing solution showcased in Nintendo’s new Wii Motion Plus Accessory”. InvenSense, Inc. http://invensense.com/mems/gyro/documents/articles/071508.html

InvenSense 2009. “MEMS Gyroscopes for Gaming”. InvenSense, Inc. http://invensense.com/mems/gaming.html

Lee, SK., Buxton, W. and Smith, K.C 1985. “A multi-touch three dimensional touch-sensitive tablet”. CHI’85 Proceedings April 1985, 21-25, ACM 0-89791-149-0/85/004/0021 http://www.billbuxton.com/leebuxtonsmith.pdf

LPRS 2005. “ER400TRS Transceiver”. Low Power Radio Solutions Ltd. http://www.lprs.co.uk/datasheets/ER400TS-02.pdf
Minor 2009. “OSCemote – iPhone Application. Open sound control in the palm of your hand”. http://pixelverse.org/iphone/oscemote/
Modbus 2010. “Modbus TCP Toolkit”. Modbus Organization, Inc. http://www.modbus.org/toolkit.php
Muvium 2005. “µVMDK Muvium Device Development Kit.” http://www.muvium.com
Nintendo 2010. “Wii Controllers”. Nintendo. http://www.nintendo.com/wii/what/controllers
Parallax Inc. 2006. “Javelin Stamp manual version 1.0a” http://www.parallax.com/Portals/0/Downloads/docs/prod/javelin/JavelinStampMan1-0.pdf
Rekdalsbakken, W. 2006. “Design and Application of a Motion Platform for a High-Speed Craft Simulator.” Proceedings of ICM 2006, IEEE 3rd International Conference on Mechatronics, 38-43, IEEE Catalog Number of Printed Proceedings: 06EX1432.
Rekdalsbakken, W. 2007. "The Use og Artificial Intelligence in Controlling a 6DOF Motion Platform." Proceedings of ECMS 2007, the 21th European Conference on Modelling and Simulation, 249-254. European Council for Modelling and Simulation, printed ISBN: 978-0-9553018-2-7, CD ISBN: 978-0-9553018-3-4.
Rekdalsbakken, W and Osen, O. L. 2009. "Teaching embedded control using concurrent Java" *Proceedings of ECMS 2009, the 23th European Conference on Modelling and Simulation*. European Council for Modelling and Simulation, printed ISBN: 978-0-9553018-8-9, CD ISBN: 978-0-9553018-9-6.
STMicroelectronics 2008. “LIS302DL MEMS motion sensor” http://www.st.com/stonline/products/literature/ds/12726.pdf
WiiBrew 2009. “WiiMote Accelerometer”. http://wiibrew.org/wiki/Wiimote#Accelerometer
WiiuseJ 2009. “WiiuseJ, Java API for Wiimotes” http://code.google.com/p/wiiusej/

AUTHOR BIBLIOGRAPHY

OTTAR L. OSEN is MSc. in Cybernetics from the Norwegian Institute of Technology in 1991. He is a senior instrument engineer at Offshore Simulation Centre AS, and assistant professor at Aalesund University College.

HELGE T. KRISTIANSEN is MSc. in Electrical engineering from the Norwegian Institute of Technology in 1992 and Master of Information Technology from Aalborg University in 2003. He is an assistant professor at Aalesund University College.

WEBJØRN REKDALSBAKKEN got his MSc. in Physics at The Norwegian Institute of Technology in 1977. He has been rector at Aalesund Engineering College, and is now assoc. professor and leader of the BSc. programme in Automation at Aalesund University College.