Evolution of robot programming towards Ubiquitous Computing

Slim Chtourou
Université de Sfax, Ecole Nationale d'Ingénieurs de Sfax
Computer & Embedded Systems CES laboratory
Sfax, Tunisia
Slim.chtourou@ieee.org

Nader Ben Amor
Université de Sfax, Ecole Nationale d'Ingénieurs de Sfax
Computer & Embedded Systems CES laboratory
Sfax, Tunisia
nader.benamor@enis.rnu.tn

Mohamed Jallouli
Université de Sfax, Ecole Nationale d'Ingénieurs de Sfax
Computer & Embedded Systems CES laboratory
Sfax, Tunisia
mohamed.jallouli@enis.rnu.tn

Mohamed Kharrat
University of Tokyo
Interdisciplinary Information Studies
Tokyo, Japan
Mohamed.kharrat@gmail.com

Mohamed Abid
Université de Sfax, Ecole Nationale d'Ingénieurs de Sfax
Computer & Embedded Systems CES laboratory
Sfax, Tunisia
mohamed.abid@enis.rnu.tn

Abstract— **In the near future, computers will be embedded everywhere and the Ubiquitous computing will be widely used. Ubiquitous computing requires multidisciplinary knowledge. The first step in a design is prototyping, however there is a lack for platforms that facilitates the development of Ubiquitous systems.**

One of the possible application fields of ubiquitous Computing is robotics. New approaches for programming and controlling robots in an easy way have to be created. They would reach a larger audience and allow beginners to create their own generic platform based project without needing strong computer and microcontroller programming skills.

This paper presents examples of programming robots made easy on the hardware and software level using Arduino and IOIO board.

Keywords: Ubiquitous computing, robots, Arduino, IOIO.

I. INTRODUCTION

If you ask a bunch of people a question: how many worlds exist in this universe? Answers may vary from 2, 3 a dozen of worlds… all these answers have one thing in common: there is more than just one world and they exist in parallel with our world: human world. Now imagine what it would be if a person is able to build a world just for himself! 30 years earlier, a person would laugh at these words but now, this "science-fiction" statement is becoming a reality thanks to the ubiquitous computing.

The Ubiquitous computing was introduced for the first time in 1988 by Mark Weiser. He put a vision of Ubicomp: a whole system that provides the right information at the correct place, in the correct time [1]. This system is like a parallel world populated by electronic devices. These devices communicate with each other, work smartly with and collaborate with other devices to achieve their ultimate objective: make humans life as easier and as comfortable as possible. Intelligent computers can adapt themselves to the human environment and autonomously choose the best configuration. These computers are expected to be integrated into everyday activities and objects such as furniture's tables, chairs, beds and even clothes. This opens perspectives for new applications such as smart house, augmented reality products, autonomous and intelligent robots…

II. ASPECTS OF UBIQUITOUS COMPUTING

A. Natural interface:

The most popular interaction method between human and machines is mouse and keyboard. These interfaces are classic and old. They don't meet with the Ubicomp requirements that aim to achieve the most effective technology by being invisible to the user. Therefore Ubiquitous computing suggests new paradigms of interaction, due to the widespread access to internet and proliferation of wirelessly interconnected devices.

Humans use different means to communicate between each other: speech, writing, gestures, eye contact… To reach the perfect Ubiquitous system, it is these means that should be used as an input and are called natural interfaces. An application that handles natural interfaces supports raw input such as voice, video, writing, sensor's input [2]…

B. Context awareness:

Devices using Ubiquitous computing should possess the ability to adapt itself automatically to the human needs and basing on the information collected from physical and computational environment. For that the Ubiquitous system has to gather knowledge to identify the actual context and decide what to do. [3]

C. Automated capture and access:

Humans can’t take note or record everything surrounding them, sometimes they miss important information. Ubiquitous computing provides tools that will record this data for them. Ubicomp applications can automate the capture of live experiences and provide access to these experiences later. [4]

III. OVERVIEW OF EXISTING PROJECTS

Researches in Ubiquitous computing focus on leveraging a higher abstraction level to make technology more accessible to normal people. There are various researches and projects that have been done in the robotics field that allows a user to build and program his first robot quite easily? In this paper we present some of these projects

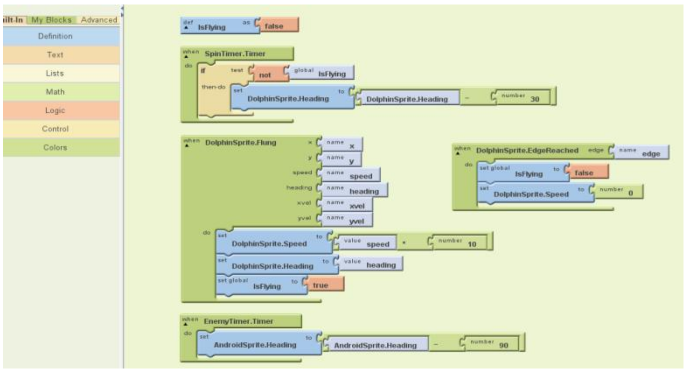
A. Massage Chair:

The Japanese home electronics released in the last years a new line of massage chair [5], raising a new competition for the integration of electronics inside the home appliance. However most of the professional involved with furniture design are not specialists which prevents them from entering this new market.

Figure 1. Panasonic EP-MS40 massage Sofa Chair [5]

B. AppInventor:

AppInventor is an online open source platform created by MIT [6] that simplifies the Android programming process to the concept of interconnected blocks with values to configure. There is no Java code to write, no need to take Android constraints into account and the design and script interface are user-friendly. With this platform, even kids are able to create interesting projects.



```
def IsFlying as false

when SpinTimer.Timer
do if test not global IsFlying
   then-do set DolphinSprite.Heading to DolphinSprite.Heading - number 30

when DolphinSprite.Flung x name x
                         y name y
                         speed name speed
                         heading name heading
                         xvel name xvel
                         yvel name yvel
do set DolphinSprite.Speed to value speed * number 10
   set DolphinSprite.Heading to value heading
   set global IsFlying to true

when DolphinSprite.EdgeReached edge name edge
do set global IsFlying to false
   set DolphinSprite.Speed to number 0

when EnemyTimer.Timer
do set AndroidSprite.Heading to AndroidSprite.Heading - number 90
```

Figure 2. AppInventor script interface

C. Upverter:

Many persons have trouble building their circuit whether for lack of software licences or materials. Upverter is an open source online platform that allows a user to design a circuit, create the schematic layout as illustrated in Figure 3.

After paying for the materials, construction and delivery fees, the circuit board will be created and delivered to the user later on. [7]

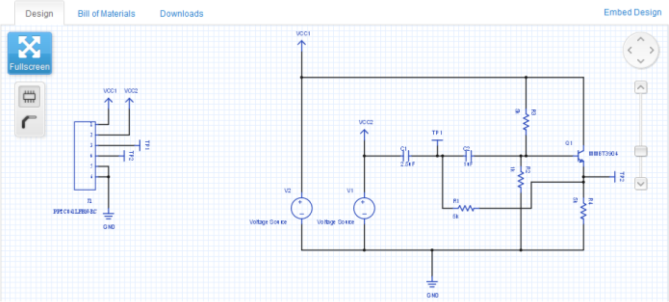


Figure 3. Sample upverter project

IV. CHALLENGES IN ROBOT PROGRAMMING

The ultimate objective of ubiquitous computing is to make technology fade in the background by making its use as easy as possible. With the rising interest in robotics, several efforts have been made to apply Ubiquitous computing and simplify the process of developing robot based projects. However when developing these projects, a user faces several constraints and challenges.

In this section we present some of the challenges faced in robotics field when developing program for robots.

A. Wires

In order for electronic devices to work, it is necessary to use cables for different purposes. This may lead to a cable congestion problem. It is not always possible to use cables, for example when exploring hostile or unknown environments.

The wireless technology has appeared, to solve wiring connection problems, allowing people to break their bonds with cables and have easier access to information stored anywhere from any place.

B. Mobility

To program a robot, we need to write a code first, most of available IDEs are for computers, which means that a user has always to rely on a computer in order to create programs. However, recently, we notice more and more a shift especially among general public toward the purchase and use of Smartphones and tablets. Which lacks the presence of IDEs for hardware programming on these devices.

C. Language complexity

Most robots are controlled through a microcontroller or a DSP. To program these devices, a developer uses the C language and more recently can use also Java. The problem is that these languages are too complicated for a user who has no or little programming skills, this limits the robotics arana to a little community of highly qualified developers where there are no place for beginners.

Context of work:

In this paper we present the concept of a more general system that can be used at large scale, in various domains and by different users. As a global vision, the ultimate objective is to create a system that allows the user to program an electronic system through an intuitive interface, without prior programming skills. All complex tasks are done in the background, what's left for the user is very simple steps to take and the program is done.

The idea came after seeing the big gap between traditional industry and new technology industry. Craftsmen for example have almost no knowledge in the programming domain and thus, they can't integrate embedded system into their furniture. On the other side, the electronic industries are not able to identify business opportunities and needs in other industries.

In this context we try to reduce this gap and create a product that allows the combination between these two types of industries. It aims to generate a better product, taking profit of recent technological progress, all in making it accessible for users with limited knowledge about electronics and computers.

V. Robot Programming Examples

In this paper, we present a proof of concept and we will focus on the robotics area and illustarte examples of robot programming made easy both on hardware and software level

A. *Wireless programming platform*

Wireless technology has solved cable congestion problem, allowing people to break their bonds with cables and have easier access to information. In this section we present a wireless programming system that allows a microcontroller to be programmed from a distance without plugging it to an external programmer.

The wireless programming system is illustrated in figure 4.

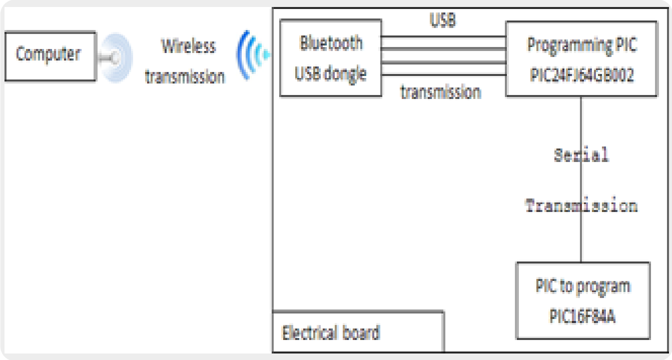


Figure 4. Overview of the wireless device programmer with PIC to program

The program file (.hex file for a PIC) will be transmitted from the computer via wireless transmission to the wireless module in the taargeted system embedded in the electrical board. Once received it the data will be transmitted to a programming microcontroller that will be in charge of handling the data to send to the main microcontroller in the board to program it. In the example in Figure 4, we use Bluetooth for wireless transmission and PIC as microcontrollers.

So once the programming file is sent over Bluetooth, it will be first received by the programming PIC. Then the file is sent byte by byte over serial transmission to the main PIC to program. The microcontroller used in this system is PIC24FJ64, which particularely has the OTG feature which allows it interfacing USB devices to ensure the USB connectivity with the Bluetooth USB dongle.

B. *Programming robot with Arduino*

B.1. *Introduction*

Arduino is an open-source prototyping platform that allows the user to build various circuits and program them easily, there are various types of Arduino boards, in this paper we'll see the Arduino BT which integrates Bluetooth connection allowing therefore programming the board wirelessly.

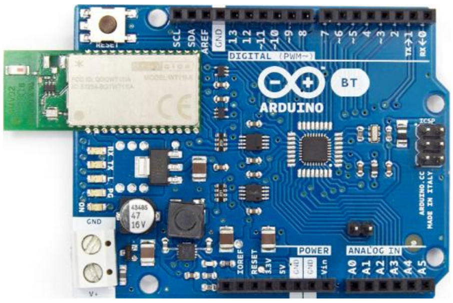


Figure 5. Arduino board (front view) [8]

B.2. *Wireless robot programming*

In this section we'll see how to control a robot using Arduino and how to reprogram the robot behavior using Bluetooth connection. Our system will be a car robot that can go forward and backward and change its motor speed. The car motor is a small motor that can be powered by 5V power supply.

The figure 6 shows the schematic used.

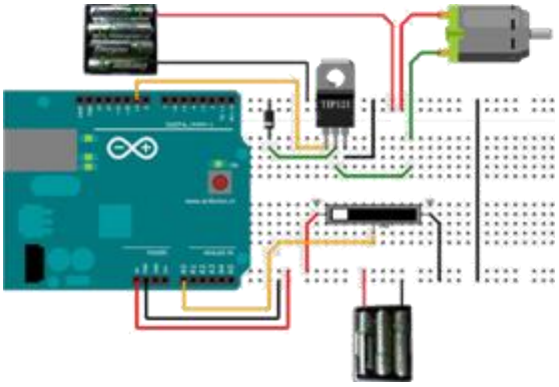


Figure 6. Robot motor speed control's schematic

The figure 7 shows the developed system:

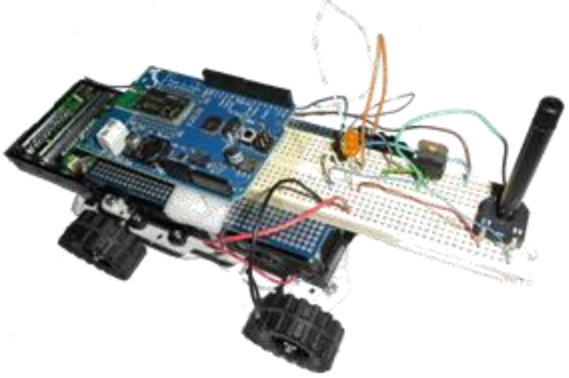


Figure 7. Robot + wireless programming with Arduino

Now after building the hardware part, we move to the software side. To illustrate changes in Arduino BT board's program, we begin by a sample program that consists of a blinking LED as described in the following code:

```
void setup() {
pinMode(9, OUTPUT); }  // set the LED pin as output
void loop() {
digitalWrite(9, HIGH);// set the LED on
delay(1000);          // wait for a second
digitalWrite(9, LOW);  // set the LED off
delay(1000); }        // wait for a second
```

The program consists a LED is connected to Arduino's digital pin no°9 which is configured as an output pin, it will be turned on for a second and then turned off for another second, this cycle will repeat itself in infinite loop.

A second application consists of controlling the robot motor speed by reading a potentiometer output

```
int dc_speed = 0; // value read from  potentiometer
void setup() {
pinMode(9, OUTPUT); }// set transistor pin as output
void loop() {
dc_speed = analogRead(0)/4; // read potentiometer
analogWrite(9, dc_speed); } //control the transistor
```

We intend to store the speed value into a variable called dc_speed, for that we need first to initialize this variable to 0.

We will use one of Arduino pins to deliver the speed value to the transistor, so we need to configure it as output
The potentiometer output is connected to one of Arduino BT's analog pins. Analog inputs provide a 10 bits resolution, so the obtained values range in [0..1023], However the board's digital pin provides an 8 bits PWM signal when using the analogWrite function, so the output value ranges in the interval [0..255].

The input and output values are not compatible because they don't have the same interval, to remedy to this problem we divide the value read from the potentiometer by 4. This way, the obtained value will be converted to adapt it to the [0.255] interval, the same as the output pin and can therefore be transmitted correctly to the transistor.

As a caution, before uploading this program, we need to disconnect the power stage external power supply and set the potentiometer output voltage to 0 to avoid damaging the board.

C. Programming robot with IOIO

C.1. Intorduction

We present in this section another example of robot wireless programming using a different board which is the IOIO.

IOIO is another open source board designed to add advanced hardware I/O capabilities to a PC or android application [9]. The board used here is IOIO OTG which allows it to interface USB devices connected to it. With this feature, it is possible to enable wireless transmission by inserting a Bluetooth USB dongle, allowing therefore to program the board wirelessly whether from a PC or a Smartphone.

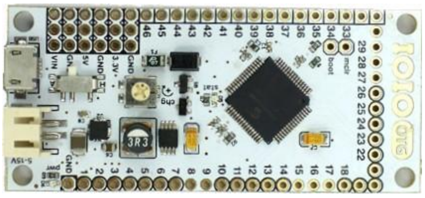


Figure 8. IOIO board (front view) [9]

C.2. Wireless robot control IOIO

In this paper, we'll illustrate the example of controlling a robot car using IOIO board as shown in figure 9

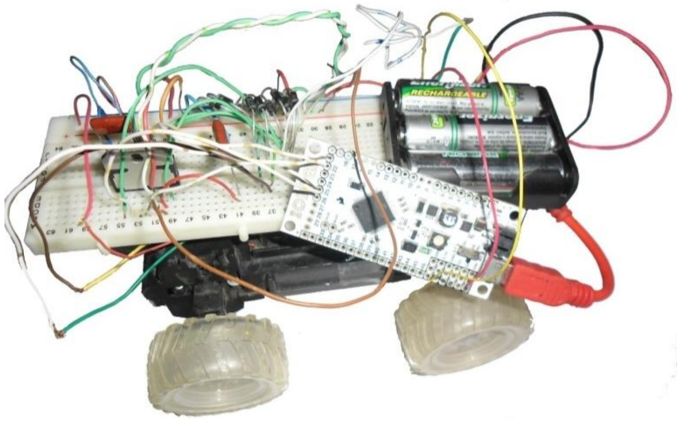


Figure 9. Robot + wireless remote circuit with IOIO

The robot has 2 motors: one for moving forward/backward, and the other to control direction, to control the robot motors, we will use a dual H-bridge motor: L298N

The L298 will power motors in the desired directions according to logic signal connected to the IOIO board.

The IOIO receives the signal from a Smartphone that has the application created with Eclipse and installed, by synchronizing Bluetooth connection and depending on user's actions.

The figure 10 shows the system's schematic.

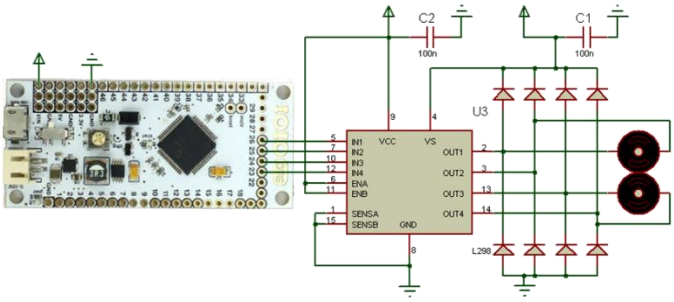


Figure 10. Robot wireless remote control with IOIO schematic

VI. CONCLUSION

The ultimate objective of Ubiquitous Computing is to set the best environment for humans and simplify doing things.

In this paper we present a proof of concept of Ubiquitous Computing applied in the robotics fields in the context of making robot programming simpler. We introduced wireless technology as a tool to program devices wirelessly, then we present some hardware solutions that ease the process of prototyping and creating robot based projects

There are several ubiquitous systems but they present the drawback of being too specific and they mainly target hardware. There is still no software oriented systems that make the robot programming task possible without needing a strong programming skills.

REFERENCES

[1] Weiser, M. 1991. *The computer for the 21st century*. Sci. Am. 265, 3 (Sept.), 94–104

[2] Gregory D. Abowd and Elizabeth D. Mynatt, *charting past, present and future research in ubiquitous computing*, ACM Transactions on Computer-Human Interaction, Vol. 7, No. 1, March 2000, Page 32

[3] Gregory D. Abowd and Elizabeth D. Mynatt, *charting past, present and future research in ubiquitous computing*, ACM Transactions on Computer-Human Interaction, Vol. 7, No. 1, March 2000, Page 35

[4] Gregory D. Abowd and Elizabeth D. Mynatt, *charting past, present and future research in ubiquitous computing*, ACM Transactions on Computer-Human Interaction, Vol. 7, No. 1, March 2000, Page 39

[5] Decisive Magazine, Panasonic EP-MS40 Massage Sofa Chair, www.decisivemagazine.com/panasonic-ep-ms40-massage-sofa-chair, 19 June 2012 [7 Oct 2013]

[6] John Sonmez, "introduction to App Inventor" *in learn to build apps with App Inventor*, pluralsight hardcore developer training, 2013, slide. 3

[7] Upverter, features, www.upverter.com, 1 October 2013 [7 Oct 2013]

[8] Arduino official site, home page, www.arduino.cc, 13 February 2011 [7 Oct 2013]

[9] IOIO GitHub repository,ytai/ioio wiki, www. github.com/ytai/ioio/wiki, 23 March 2013 [7 Oct 2013]