



ANDROID CONTROLLED SPY-ROBOT

Prof. Y. M. Naik, Chiranjivi. M. Deshpande, Ravija.R. Shah, Rashmi. R. Kulkarni
Department of Computer Science and Engineering,
Hirasugar Institute of Technology Nidasoshi-591236
Dist: Belgaum Karnataka, INDIA

ABSTRACT: Now days we can see nearly everything which was once controlled by human being are being automated using machines and electronics circuits. In this project we are mainly concentrating on the use of a cell-phone to run an external hardware device. Previously it was not possible to do so; just because of the emergence of android it became possible. Smartphones are becoming each time more powerful and equipped with new accessories that are important for robots. In this project we make use of 2 smartphones to form a spy robot. One cellphone acts as a controller and other as the spy. Hence the system is a low cost controller useful in all the fields of life.

KEYWORDS: ADB (Android Debug Bridge); NDK (Native Development Kit); API (Application Program Interface); MCU (Microcontroller Unit); APK(Application Package Kit); GPS (Global Pointing System); USB (Universal Serial Bus); SDK (Software Development Kit); UART (Universal Asynchronous Receiver and Transmitter, RX (Receiver); TX (Transmitter); SBUF (special buffer); IR (Infrared)

I. INTRODUCTION

Android phones have a huge potential as low cost robot controllers. Android programming is done in java which makes it very attractive to program because of familiarity to java. They have built in GPS compasses and cameras, Bluetooth and high end processors running at an average of 500Mhz. Hence the project establishes a bridge between the processor Android in the Smartphone and the microcontroller in the robot. The external interface is totally controlled by means of the mobile phone. By being able to use the android phone as a controller for the robots we form a cost efficient, powerful and extensible basis robot. This is even useful to give a richer and cohesive experience to the user.

The paper mainly concentrates on human gestures to control the hardware device. The device i.e. the android mobile phone will recognize the gesture and pass on the information to the microcontroller which will make the device (robot) move accordingly. Moreover the camera will capture pictures all over the movement (via the cell phone placed in the robot) and those pictures will be sent to the email-id (registered one). A number of techniques are available for robot control however they have difficult user interface and limited to certain environments. However what is really lacking is a convincing way to interaction friendly to all.

II. RELATED WORK (SURVEY)

Why robots to be controlled by gesture? Robots are artificial objects capable of perceiving and acting in the physical world. Nowadays, due to technological advances, robots have become more powerful, intelligent and available to be used in all aspects in all fields. With the price of robots dropping they are being introduced into numerous and various areas and applications, ranging from common domains to highly complex and uncertain environments and from relatively safe to highly dangerous environments.

In these application scenarios the robots very often necessitate the human operator's expertise and intelligence which are communicated to them in different forms. Therefore, contemporary robotics demands interface devices that are accurate, natural and easy to use by everyone, ubiquitous and inexpensive. As robotics becomes more sophisticated, the users of the apparatus are likely to become less skilled, thus imposing additional importance on the easiness of use of the interface device.

Comparison to other technologies: Various forms of human-robot interface have been proposed so far. Common solutions for interacting with robots use a Graphical User Interface (GUI) where the user views commands and often introduces commands through touch screen, joystick, mouse and/or keyboards of different kinds. Frequently these devices are local and wired to the robot. Moreover, with these kinds of interface the operator is

constantly giving direct low-level commands, which is limiting in applications where more abstract instructions and greater robot autonomy are required.

To make robot control as-autonomous-as possible, other high-level control approaches, were introduced, from which we have chosen hand gesture.

Some high level approaches include smart brain-computer interfaces are for effective human-exoskeleton interaction in stroke rehabilitation tasks wherein A natural interface based on body language movements processed via a 3D camera is described .There are many more high level interfaces available.

However, all these high-level interfaces are equivocal in many cases and lack robustness for practical use. Therefore, we have considered gesture as a powerful option for high-level robot control. In our work we are proposing a pointing interface based on human gestures using a mobile phone accelerometer for interaction with robots.

Why embedded c code? : When designing software for a smaller embedded system with the 8051, it is very common place to develop the entire product using assembly code. With many projects, this is a feasible approach since the amount of code that must be generated is typically less than 8 kilobytes and is relatively simple in nature. The trouble with projects done with assembly code can be that they can be difficult to read and maintain. A program written in C is easier to read than an assembly program.

Since a C program possesses greater structure, it is easier to understand and maintain. A high order language such as C allows a developer to write code, which resembles a human's thought process more closely than does the equivalent assembly code.

The programmer does not have to be intimately familiar with the architecture of the processor. This means that someone new to a given processor can get a project up and running quicker, since the internals and organization of the target processor do not have to be learned. Additionally, code developed in C will be more portable to other systems than code developed in assembly.

EXISTING SYSTEM: The existing system consists of the use of gloves to give human interaction to the machine (for gesture interpretation) however this technique is not user friendly and requires a lot of effort to get the output. Next technique is to use a separate controller. Again engineers have to give separate training to the persons in charge for controlling. Separate sessions are required to do so which is again not time and cost efficient. Again separate controller rooms are required in case of army and military spying operations which are very difficult to manage. Next command line interfaces and graphical user interfaces can be used for interaction which is not feasible.

III. PROPOSED WORK

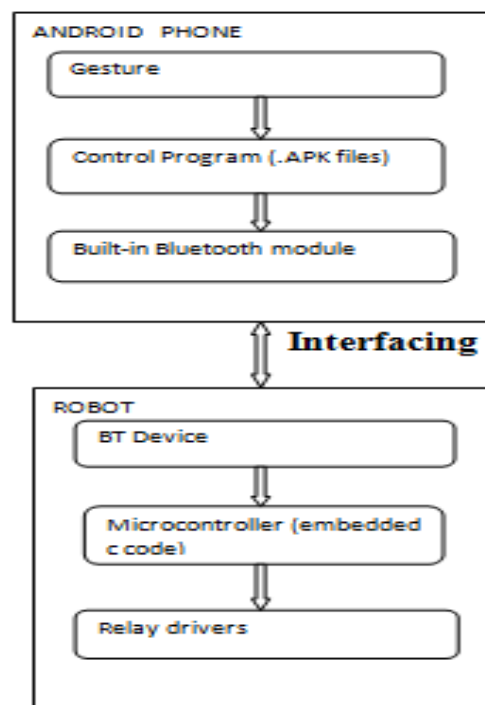
The proposed work runs over all these disadvantages of the existing systems and develops an android based spy-robot which is

- **Powerful:** Through the high processor performance of the android phone robotic application will get high power.
- **Easy to use:** Android is open and provides open interfaces for peripherals. Hence as it is friendly to all to make use of a cell-phone. It becomes easy to use with no separate remote or controller requirements.
- **Affordable:** By using a device that most people already own it becomes very cost effective. Apart from that as the API's are open they are free to use.

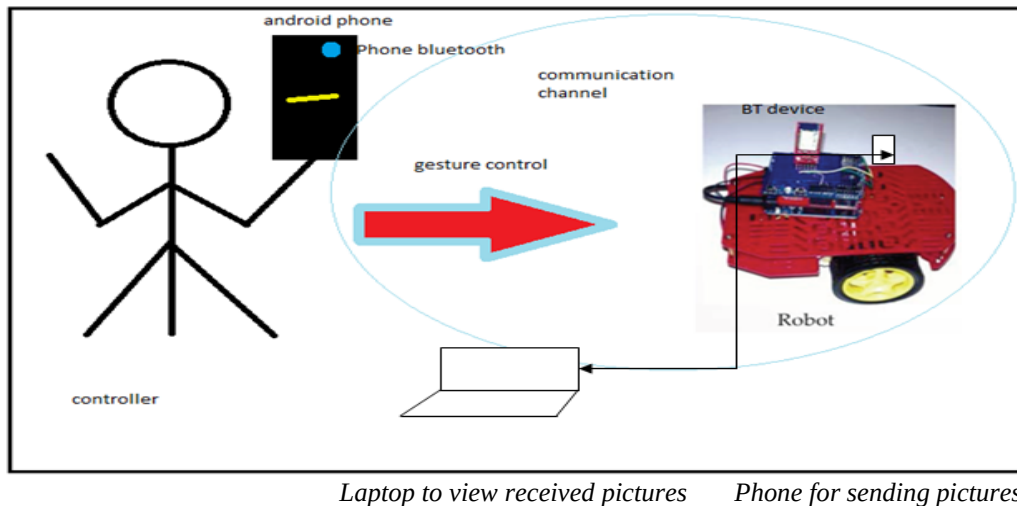
SYSTEM OVERVIEW: The system consists of the following three modules

- (a) **ROBOT:** The robot consists of the use of a microcontroller 8051 that has the embedded c code dumped into it. Again it consists of an IR sensor for obstacle detection.
- (b) **CONTROLLER:** The controller has got an android cell-phone. The cell phone is connected to the Bluetooth in the robot. This Bluetooth establishes a communication channel between the controller and the robot. The '.apk' file i.e. the target file is stored into the cell-phone. When this is run then the GUI i.e. the screen on the cell phone occurs and the user gives gestures. These gestures are converted into analog signals and transmitted to the robot. Here they are converted into digital signals and the microcontroller code starts executing and the robot moves according to the gestures.
- (c) **CAMERA:** The second android phone is placed in the robot. This phone is responsible for taking pictures. When an 'enable' message is sent to the phone the camera in the phone gets automatically

activated and starts capturing pictures at various angles. The cell phone must have internet. The sender and the receiver email id are registered in the android code in the cell phone. Hence the camera gets enabled and sends the necessary pictures to the email-id until a 'close' message is sent to the phone. Once the close message is sent the activity stops.



SNAPSHOT OF THE ENTIRE SYSTEM



IV. INTERFACING

The android code is written in java and the microcontroller code is written in embedded c. This is possible with the help of android NDK. The NDK allows us to program the hardware in our known language. Hence the linking of codes is done via the NDK.

V. CONCLUSION AND FUTURE WORK

With the available facilities and infrastructures provided we are successful in designing a cost-effective system which accomplishes the required application. Implementation of wireless technology in the project enables the handling of the robot efficiently without manual intervention. With slight modification it can also be used for lot many other different application. The proximity sensors should be in contact with metallic path thus providing an inefficient method of movement. This can be overcome by using inductive other sensors. The speed can be increased using faster motors in future.

VI. REFERENCES

- Android developers, <http://developer.android.com>.
- Kanjo E., Bacon J., Roberts D., and Landshoff P., "MobSens: Making Smart Phones Smarter," IEEE Pervasive Computing, vol. 8, no. 4, pp. 50-57, Oct 2009.
- Shaer O., Leland N., Calvillo-Gamez E. H., and Jacob R. J., "The TAC paradigm: Specifying tangible user interfaces," Personal Ubiquitous Computers", vol. 8, no. 5, pp.359-369, Sep. 2010.
- Arduino, <http://arduino.cc>.
- Weiser M., "Some computer science issues in ubiquitous computing," Comm. ACM, vol. 36, no. 7, pp. 75-84, Jul. 2013.
- Schultz J., "Ubiquitous computing goes mobile," SIGMOBILE Mob. Comput. Comm. vol. 5, no. 3, pp. 32-38, Jul. 2011.
- Williams A., Farnham S., and Counts S., "Exploring wearable ambient displays for Social awareness," in CHI '06 Extended Abstracts on Human Factors in Computing Systems, Montréal, Québec, Canada, April 22 - 27, 2008. CHI'06. ACM, New York.