

Development of Vision-Based Sensor of Smart Robot for Industrial Applications

D.Vijaya Bhavani, Shaik Meeravali

Abstract - Robotic manipulators are widely used to replace human operators in tasks that are repetitive in nature. However, there are many tasks that are non-repetitive, unpredictable, or hazardous to the human operators. Clear in gupanu clear power plant leak or exploring the extreme depths of ocean are just some examples. The most developed robot in practical use to day is the robot car mandit is seen in applications throughout the world. Robotic are used to carry out working outer space where man can not survive and also used to do work in them edical fields such as conducting experiments without exposing the researcher.

In early days, robotic manipulators have been implemented in different control techniques like mechanical control and the remote control or tele-operation. But with the advent of high performance, a new way of control using mobile has been implemented which is introduced in this project.

All the above systems are controlled by the Microcontroller. In our project we are using the popular 8-bit microcontroller AT89S52. It's a 40 pin microcontroller. The Microcontroller AT89S52 is used to control the DC motors. Two DC motors are used to drive the robot in front direction. i.e. Front. The robot is also developed to give an alert when any fire accidents occurs and to give an alert. Here we are using fire sensor and IR-pairs for altering fire sensors and for obstacle detection also.

Index Terms- Mobile robots, Navigation, Robot vision systems, Intelligent robots, Learning systems, Cooperative systems.

I INTRODUCTION

Task specification in autonomous robotics has received an increasing interest. It is today admitted that autonomous mobile robots should be designed with a minimal prior knowledge on the tasks to perform so that the robot can adapt to unpredictable situations characterizing the dynamical nature of real environments. The robots should also constitute their skills via interactions with their physical and social environment where they build up experiences from their sensory-motor interactions leading their own cognition to enact a subjective world, also called the Umwelt. In this context, Human-Robot Interactions (HRI) are thought to be a very efficient means to specify some various tasks to a robot and to catalyze its sensory-motor learning. HRI are moreover a key point for designing operational or social and interactive robot. This paper investigates the use of HRI for the learning of navigation tasks.

This paper first presents our robots and its visual system enabling to create a continuous state space. Then, we will propose a bootstrap for the PerAc architecture that enables the semi-supervised learning of a sensory-motor behavior (a visual path, a homing behavior).

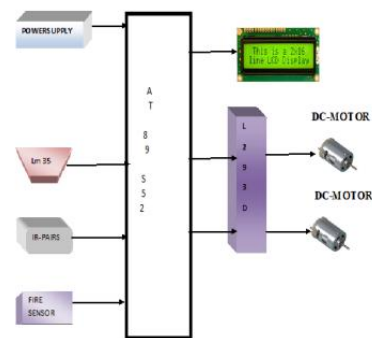
Manuscript received on September, 2013.

D.Vijaya Bhavani, M.Tech (Embedded Systems), ECE Department, RRS College of Engineering & Technology, Muttangi, Andhra Pradesh, India

Dr. Shaik Meeravali, Professor & HOD, ECE Department, RRS College of Engineering & Technology, Muttangi, Andhra Pradesh, India

The couple architecture- equations enables to adapt the partitioning of the environment to the complexity of the task. The system does not separate learning and performing phases, which are scattered in time according to the rhythm of the interaction. The system will be evaluated in a real indoor environment by means of accuracy measures between the performed trajectory and the expected behavioral attractor of the robot dynamics.

BLOCK DIAGRAM:



II MICROCONTROLLER

Microcontrollers are "embedded" inside some other device (often a consumer product) so that they can control the features or actions of the product. Another name for a microcontroller, therefore, is "embedded controller". Microcontrollers are dedicated to one task and run one specific program. The program is stored in ROM (read-only memory) and generally does not change.

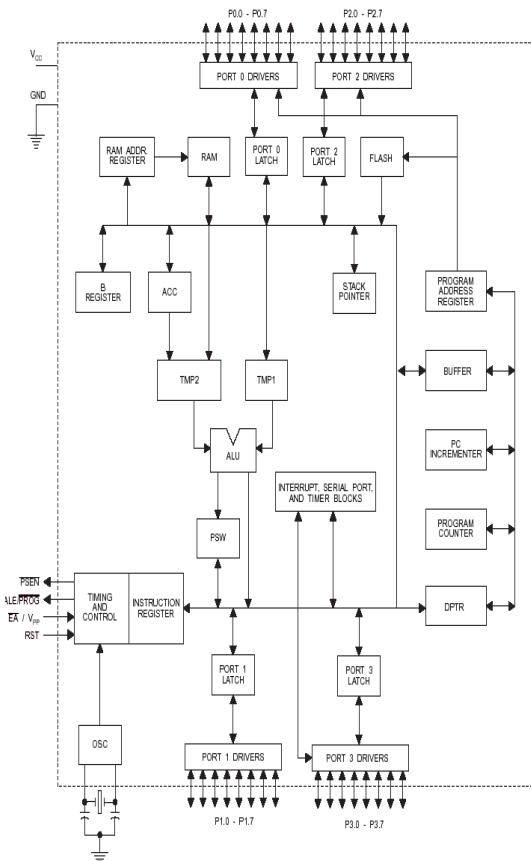
Microcontrollers are often low-power devices and have a dedicated input device and often (but not always) have a small LED or LCD display for output. A microcontroller also takes input from the device it is controlling and controls the device by sending signals to different components in the device.

For example, the microcontroller inside a TV takes input from the remote control and displays output on the TV screen. The controller controls the channel selector, the speaker system and certain adjustments on the picture tube electronics such as tint and brightness. The engine controller in a car takes input from sensors such as the oxygen and knock sensors and controls things like fuel mix and spark plug timing. A microwave oven controller takes input from a keypad, displays output on an LCD display and controls a relay that turns the microwave generator on and off. A microcontroller software is small and low cost. The components are chosen to minimize size and to be as inexpensive as possible.

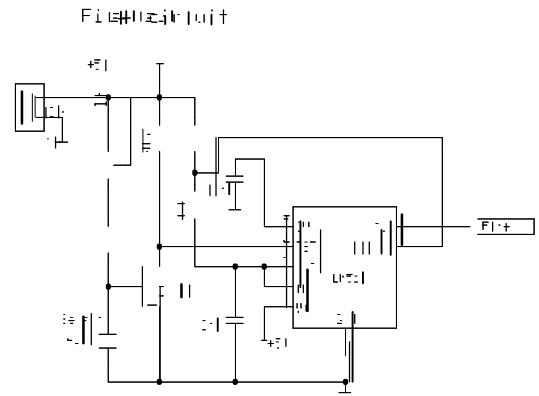
A microcontroller is often, but not always, ruggedized in some way. The microcontroller controlling a car's engine, for example, has to work in temperature extremes that a normal computer generally cannot handle. A car's microcontroller in Alaska has to work fine in -30 degree F (-34C) weather, while the same microcontroller in Nevada might be operating at 120 degrees F (49C). When you add the heat naturally generated by the engine, the temperature can go as high as 150 or 180 degrees F (65-80C) in the engine compartment. On the other hand, a microcontroller embedded inside a VCR hasn't been ruggedized at all. The actual processor used to implement a microcontroller can vary widely.

The Intel 8051 is Harvard architecture, single chip microcontroller (μC) which was developed by Intel in 1980 for use in embedded systems. The official designation for the 8051 family is MCS 51. Intel's original versions were popular in the 1980s and early 1990s, but has today largely been superseded by a vast range of faster and/or functionally enhanced 8051-compatible devices manufactured by more than 20 independent manufacturers including Atmel, Infineon Technologies (formerly Siemens AG), Maxim Integrated Products (via its Dallas Semiconductor subsidiary), NXP (formerly Philips Semiconductor), Nuvoton (formerly Winbond), ST Microelectronics, Silicon Laboratories (formerly Cygnal), Texas Instruments and Cypress Semiconductor.

Intel's original 8051 family was developed using NMOS Technology, but later versions, identified by a letter C in their name (e.g., 80C51) used CMOS technology and were less power-hungry than their NMOS predecessors. This made them more suitable for battery-powered devices.



III FIRE SENSOR CIRCUIT



IV TEMPERATURE SENSOR (LM35)

Precision Centigrade Temperature Sensor: In this project, in order to monitor the temperature continuously and compare this with the set temperature preprogrammed in the microcontroller, initially this temperature value has to be read and fed to the microcontroller. This temperature value has to be sensed. Thus a sensor has to be used and the sensor used in this project is LM35. It converts temperature value into electrical signals.

LM35 series sensors are precision integrated-circuit temperature sensors whose output voltage is linearly proportional to the Celsius temperature. The LM35 requires no external calibrations since it is internally calibrated. The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1/4° C at room temperature and ±3/4° C over a full -55 to +150° C temperature range.

The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout control circuitry especially easy. It can be supplied with single power supplies, or with plus and minus supplies. As it draws only 60μA from its supply, it has very low self-heating, less than 0.1° C in still air.

V WHAT IS INFRARED

Infrared is an energy radiation with a frequency below our eyes' sensitivity, so we cannot see it. Even though we cannot "see" sound frequencies, we know that it exists, we can listen to them.



Even though we cannot see or hear infrared, we can feel it at our skin temperature sensors. When you approach your hand to fire or warm element, you will "feel" the heat, but you can't see it. You can see the fire because it emits other types of radiation, visible to your eyes, but it also emits lots of infrared that you can only feel in your skin.



VI INFRARED IN ELECTRONICS

Infra-Red is interesting, because it is easily generated and doesn't suffer electromagnetic interference, so it is nicely used to communication and control, but it is not perfect, some other light emissions could contain infrared as well, and that can interfere in this communication. The sun is an example, since it emits a wide spectrum of radiation.

The adventure of using lots of infra-red in TV/VCR remote controls and other applications, brought infra-red diodes (emitter and receivers) at a very low cost at the market.

From now on you should think of infrared as just a "red" light. This light can mean something to the receiver, the "on or off" radiation can transmit different meanings. Lots of things can generate infrared, anything that radiates heat does, including our body, lamps, stove, oven, friction of your hands together, even the hot water at the faucet.

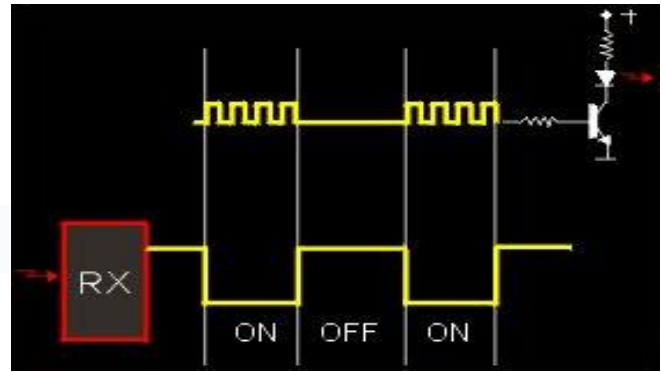
To allow good communication using infra-red, and avoid those "fake" signals, it is imperative to use a "key" that can tell the receiver what is the real data transmitted and what is fake. As an analogy, looking at the night sky you can see hundreds of stars, but you can spot easily a faraway airplane just by its flashing strobe light. That strobe light is the "key", the "coding" element that alerts us.

Similar to the airplane at the night sky, our TV room may have hundreds of tiny IR sources, our body, and the lamps around, even the hot cup of tea. A way to avoid all those other sources, is generating a key, like the flashing airplane. So, remote controls use to pulsate in infrared at a certain frequency. The IR receiver module at the TV, VCR or stereo "tunes" to this certain frequency and ignores all other IR received. The best frequency for the job is between 30 and 60 kHz, the most used is around 36 kHz.

VII IR GENERATION

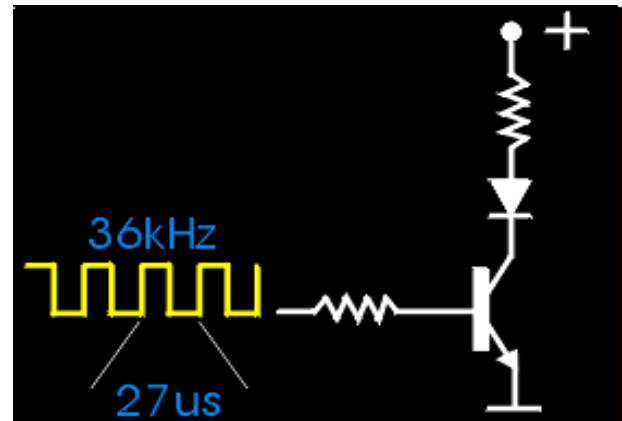
To generate 36 kHz pulsating infrared is quite easy, more difficult is to receive and identify this frequency. This is why some companies produce infrared receivers, that contain the filters, decoding circuits and the output shaper, that delivers a square wave, meaning the existence or not of the 36 kHz incoming pulsating infrared.

It means that those 3 dollars small units, have an output pin that goes high (+5V) when there is a pulsating 36 kHz infrared in front of it, and zero volts when there is not this radiation. As a square wave of approximately 27 μs (microseconds) injected at the base of a transistor, can drive an infrared LED to transmit this pulsating light wave. Upon its presence, the commercial receiver will switch its output to high level (+5V). If you can turn on and off this frequency at the transmitter, your receiver's output will indicate when the transmitter is on or off.



Those IR demodulators have inverted logic: its output, when a burst of IR is sensed, it drives its output to a low level, meaning logic level = 1.

The TV, VCR, and Audio equipment manufacturers for long use infra-red at their remote controls. To avoid a Philips remote control to change channels in a Panasonic TV, they used different codification in the infrared, even though all of them use basically the same transmitted frequency, from 36 to 50 kHz. So, all of them use a different combination of bits or how to code the transmitted data to avoid interference.



VIII ABOUT SOFTWARE

Our projects completed on schedule. Keil development tools for the 8051 Microcontroller Architecture support every level of software development from the professional applications engineer to the student just learning about embedded software development. The industry-standard Keil C Compilers, Macro Assemblers, Debuggers, Real-time Kernels, Single-board Computers, and Emulators support all 8051 derivatives and help you get.

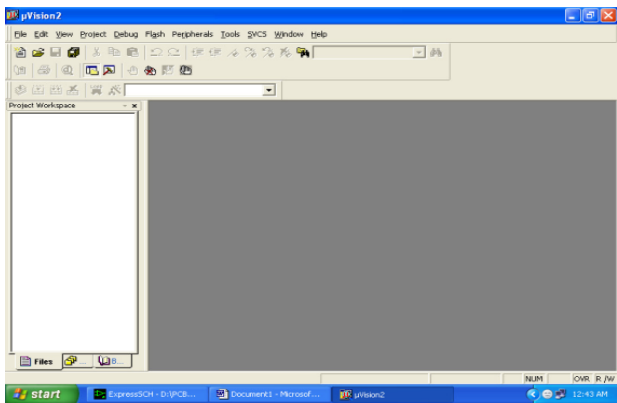
IX SIMULATION

The μVision Simulator allows you to debug programs using only your PC using simulation drivers provided by Keil and various third-party developers. A good simulation environment, like μVision, does much more than simply simulate the instruction set of a microcontroller— it simulates your entire target system including interrupts, startup code, on-chip peripherals, external signals, and I/O. This software is used for execution of microcontroller

programs. Keil development tools for the MC architectures support every level of software development from the professional applications engineer to the student just learning about embedded software development. The industry-standard Keil C compilers, macro assemblers, debuggers, real-time kernels, single-board computers and emulator support all microcontroller derivatives and help you to get more projects completed on schedule. The Keil software development tools are designed to solve the complex problems facing embedded software developers.

When starting a new project, simply select the microcontroller you use, the device database and the µVision IDE sets all compiler, assembler, linker, and memory options for you. Numerous example programs are included to help you get started with the most popular embedded AVR devices. The Keil µVision debugger accurately simulates on-chip peripherals (PC, CAN, UART, SPI, Interrupts, I/O ports, A/D converter, D/A converter and PWM modules) of your AVR device. Simulation helps you understand hardware configurations and avoid time wasted on setup problems. Additionally, with simulation, you can write and test applications before target hardware is available.

When you are ready to begin testing your software application with target hardware, use the MON51, MON390, MONADI, or flash MON51 target monitors, the ISD51 In-System Debugger, or the ULINK USB-JTAG adapter to download and test program code on your target system. Click on the Keil µVision Icon on Desktop. The following figure will appear:



X CONCLUSION

In early days, robotic manipulators have been implemented in different control techniques like mechanical control and the remote control or Tele-operations. But with the advent of high performance, a new way of robotic control with detection of an obstacle, fire and temperature values can also be successfully detected.

REFERENCES

1. Water Heating Base on Multithreading, Measurement and Control
2. Technique, Vol. 28, No. 8, pp. 79-81, 2009 (in Chinese).
3. M. Gianluigi, G. Italian, GSM and GPRS performance of IPSEC
4. Data communications, J. Ascenso et al. (eds.), e-Business
5. Telecommunication Networks, pp. 125-133, 2006
6. Management system, Lecture Notes in Computer Science, Vol.
7. M. Weske, Object-oriented design of a flexible workflow
8. 1475, Advances in Databases and Information Systems, pp. 119, 1998.

9. D. Scarpazza, P. Raghavan, D. Novo, F. Catthoor, and Diederik
10. Verkest, Software simultaneous multi-threading, a technique to exploit task-level parallelism to improve instruction- and data-level.

AUTHORS PROFILE



D Vijaya Bhavani, has done B.Tech in Pragati Engineering College in Electronic & Communication Engineering and now currently pursuing Masters of Technology in RRS College of Engineering and Technology in Embedded Systems.



Dr. Shaik Meeravali, is currently working as Professor and Head of the Department, ECE, in RRS College of Engineering and Technology.

