

Development of an Intelligent Assistant Robot based on Embedded RTOS

Fengzhi Dai

College of Electronic Information and Automation, Tianjin University of Science and Technology, Hexiqu Dagunanlu
1038, Tianjin, 300222, China, daifz@tust.edu.cn

Yuan Li, Guodong You

College of Electronic Information and Automation, Tianjin University of Science and Technology, Hexiqu Dagunanlu
1038, Tianjin, 300222, China, yuanli12139@gmail.com

Abstract

The paper focuses on the development of a wheeled intelligent robot that could be applied in hospitals or nursing homes for medical assistance. It includes two microcontrollers, MC9S12XS128 and STC89C52. The program is basically achieved on CodeWarrior 5.0, Keil μ Vision4 with the OpenCV library, porting μ C/OS-II real-time operating system to MC9S12XS128 in order to improve stability and flexibility. The robot has functions of infrared remote control, fuzzy rule-based path tracking, obstacle avoidance, color recognition and object grasping.

Keywords: Assistant robot; Microcontroller; Embedded system; μ C/OS-II.

1. Introduction

Robot and its correlation techniques have obtained a significant progress in recent years.¹ For software, the embedded real-time operating system (RTOS) has been applied into plenty of intelligent devices such as the mobile phone because of its superior portability, stability, and efficiency.

In spite of many types of electric scooters and wheelchairs have been sold in market and used in hospitals and nursing homes, most of them have simple and single functions without adopting advanced intelligent control approaches.^{2,3} For this reason, introducing the embedded RTOS to solve this problem is necessary. The versatile intelligent robot for future medical assistance, which is shown in Fig.1, was implemented in the research of this paper.

The wheeled robot has a series of functions including manual remote control, independent path tracking with

obstacle avoidance, and fundamental pattern recognition. The function of elevating under remote-control mode greatly increases the range that the manipulator can reach to.⁴

But most importantly, the application of μ C/OS-II embedded RTOS allows all the functions of the robot being coordinated in a more flexible and reasonable

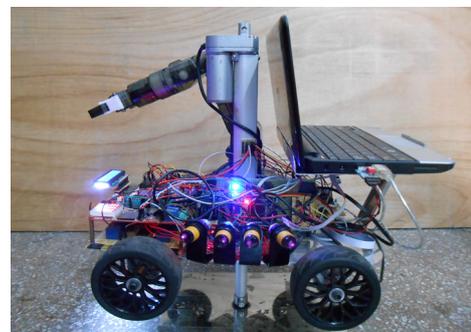


Fig. 1. Appearance of the robot

way. It deserves to believe that this type of wheeled robot will make great change and improvement for quality of life as long as it can be applied in hospitals, nursing homes or other civilian fields.

2. Hardware and Principal Functions

Two microcontrollers MC9S12XS128 (XS128) and STC89C52 (C52), cooperated with Dell Inspiron Mini10 laptop, are adopted for accomplishing all the functions of the robot.

The XS128 (ported $\mu\text{C}/\text{OS-II}$ embedded RTOS) is in charge of the driving wheels, elevating, path tracking, obstacle avoidance and the robot arm. The C52 decodes the infrared signal of remote control through the external interruption, and controls LCD display by parallel I/O ports and voice broadcast by universal asynchronous receiver/transmitter.

An interactive interface on the laptop based on the OpenCV library is designed for the real-time image capture, processing and color recognition. Two microcontrollers are communicated by their I/O ports. The laptop sends the result of color recognition by the serial port communication to XS128.

The motors of wheels and elevating platform are driven by two MC33886 and one L298N respectively; infrared remote control is achieved by HS0038 infrared receiver linked to C52 so as to control the motion of the robot and switching between the automatic and manual mode; a gray scale sensor with seven LEDs detects the path on the ground for path tracking function; five obstacle-avoiding sensors installed on the head and left edge enable the robot to avoid obstacles while tracking the path.

The robot arm owns five degrees of freedom driven by two ULN2003A and ten relays; the LCD1602 (installed on the head of the robot) and the speech synthesis module (based on SYN6288) can display and broadcast the robot's running status in real time.

3. Porting $\mu\text{C}/\text{OS-II}$ to MC9S12XS128

The main program of XS128 is compiled and debugged upon CodeWarrior 5.0 IDE. The robot's functions of remote control, path tracking, obstacle avoidance and color recognition can be managed flexibly and

efficiently under the environment of $\mu\text{C}/\text{OS-II}$ embedded RTOS. But before that, $\mu\text{C}/\text{OS-II}$ should be successfully ported to the microcontroller XS128.

CPU12X has a set of registers including a 16-bit program counter (PC), a 16-bit stack pointer register (SP), a 16-bit accumulator D, 16-bit index registers X and Y, a 16-bit condition code register, a global page (GPAGE) register and specific resource page (RPAGE, EPAGE and PPAGE) registers. For responding interruptions, the hardware of XS128 will automatically push the core registers into the stack whose address grows downward.⁵

Among $\mu\text{C}/\text{OS-II}$ files, definition of the above hardware information including data types, stack data type, stack's growth direction, tasks' scheduling function OS_TASK_SW and macro definition of critical section of code in the OS_CPU.H file; porting four assembly functions including OSStartHighRdy, OSCtxSw, OSIntCtxSw and OSTickISR in the OS_CPU_A.ASM file, and six functions including OSTaskStkInit, OSTaskCreateHook, OSTaskDelHook, OSTaskSwHook, OSTaskStatHook and OSTaskTickHook in the OS_CPU_C.C file.⁶

After porting three files mentioned above, XS128's RTI module is used to make a system clock for the RTOS. The system bus frequency is set to 32MHz. And the related registers of RTI are set and the time-tick interrupt vector is added to the file TBDML_linker.prm.

Finally, adding all header files into INCLUDES.H and finishing configuration of the file OS_CFG.H, retaining the functions about task and message mailbox. So far, the porting of $\mu\text{C}/\text{OS-II}$ is accomplished and the system is ready for programming all the needed task functions in the file MAIN.C.

4. Control Program based on $\mu\text{C}/\text{OS-II}$

The embedded RTOS can simultaneously run more than one task. Each program of the robot's function can be regarded as a task (programming in the main file MAIN.C). And for the wheeled robot, there are four tasks needed to run in the $\mu\text{C}/\text{OS-II}$ environment. According to their priorities from high to low, these four tasks are: color recognition whose priority is 6, obstacle avoidance (priority is 7), remote control (priority is 8), and the task of path tracking whose priority is 9.

Fig. 2 is the state transition diagram of the robot's tasks.

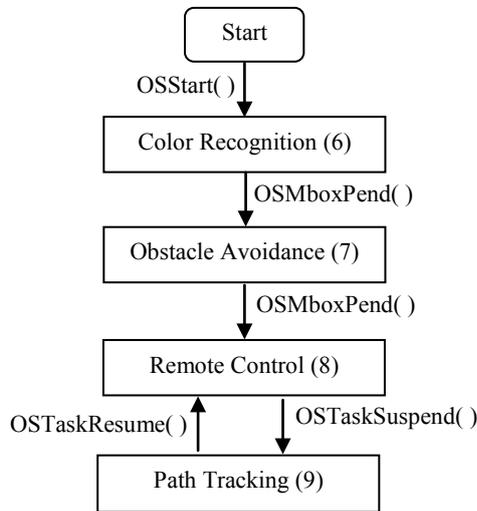


Fig. 2. The state transition of the robot's tasks

4.1. Color recognition

Red/Blue color recognition is achieved by the upper computer. Fig.3 shows the result of selecting the red color. The program is compiled on Visual C++ 6.0 with MFC and the OpenCV library. The digital camera collects HSV value of the objects within its field of view, and the computer calculates the area of the objects that match the HSV range of red or blue. The result of recognition will be sent to XS128 by the serial port for further processing.



Fig. 3. Upper computer for color recognition

Firstly, it will call the function `msg2 = (INT8U*) OSMboxPend(ExMbox2, 0, &err2)` to read the message from mailbox "ExMbox2". Then the system determines whether the message is the task that needs or not before executing the following program. If the message suggests that the found object has the selected color red or blue, the system will control the manipulator to grasp the target with that kind of color.

4.2. Obstacle avoidance

The robot is designed to automatically bypass obstacles under the path tracking mode.⁴

When the task begins, it is suspended by calling the function `msg1 = (INT8U*) OSMboxPend(ExMbox1, 0, &err1)` to wait the message from mailbox "ExMbox1" of path tracking task. Since the task of remote control or path tracking is running, the task of obstacle avoidance will not be woken up until there is obstacle detected in front of the robot.

4.3. Remote control

This function is achieved by communication protocol NEC, namely detecting duration of high level at the receiver and using the time interval between pulses to differentiate the logic "0" and "1".

An external interrupt service subroutine is designed in C52 for decoding the infrared signal. Each time after decoding, the subroutine will return an array storing the result of decoding. Then C52 will give different commands according to different values in the array to relative modules to implement certain functions, such as LCD display and voice broadcast.

C52 can also send certain remote control commands to XS128 through I/O ports of the two microcontrollers. Once XS128 receives the command of switching the operational modes from C52 by remote control, the function `OSTaskSuspend(OS_PRIO_SELF)` is called to suspend the current task and make a task switching.

Thereby, the task of path tracking (whose priority is the lowest) will acquire the access to CPU, and the robot switches to the path tracking mode.

4.4. Path tracking

The seven-channel gray scale sensors can output analog voltage signals ranging from 1.2V to 4.0V. Through the ATD channel of XS128, these analog signals are transformed to the digital signals. If the digital value is greater than the threshold, it means that the corresponding channel of gray scale sensor has detected the path. In order to control the robot more smoothly, the fuzzy control algorithm is adopted.

The state of path tracking task can also switch back to the remote control mode once receiving the switching command from C52. But in this condition, the current path tracking task should be re-suspended by calling the function OSTaskResume(Task3Prio).

Here, Task3Prio = 8, which signifies the priority of the remote control task. Since at this time the suspended task of remote control resumes into ready state, and its priority is higher than the task of path tracking, the path tracking task is instantly deprived of its access to CPU, making the running mode of the robot switch back to the remote control mode.

Also, the path tracking task can fulfill duty of sending messages to activate the tasks of obstacle avoidance and color recognition by the mailbox functions of $\mu\text{C}/\text{OS-II}$.

Once the obstacle-avoiding sensor detects the obstacle or XS128's SCI module has received the color information from the laptop, the function OSMboxPost (ExMbox1, (void*) &msg) or OSMboxPost (ExMbox2, (void*) &ch) is called for sending different messages.

5. Conclusion

This paper combines the traditional robotic technology with the embedded RTOS, designing an intelligent wheeled robot for medical assistance. After practical debugging, the robot can implement many functions like the remote control, path tracking, obstacle avoidance, color recognition, and real-time LCD display and voice broadcast, all of which can be switched flexibly by the application of $\mu\text{C}/\text{OS-II}$ embedded RTOS. Fig.4 and Fig.5 show the robot tracking the black path and avoiding obstacle.

As shown in Fig.6, the robot can also help users grasp the object of certain color by using the manipulator and the mounted digital camera. This kind of robot could be

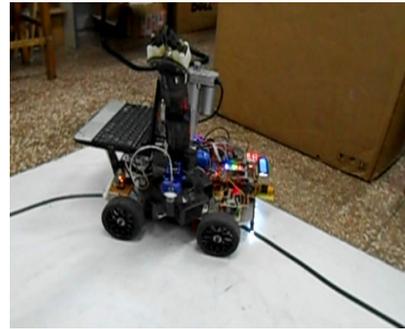


Fig. 4. The robot runs under path tracking mode

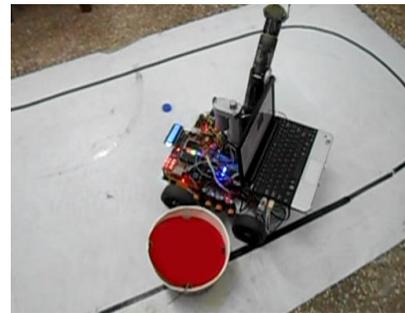


Fig. 5. The robot bypasses obstacle



Fig. 6. The robot recognizes and grasps target object

applied into hospitals or nursing homes in future to take care of old people or patients with their life and lower the burden of nurses.

Acknowledgements

The research is partly supported by the Scientific Research Foundation of Tianjin University of Science and Technology (20130123), and the Open Foundation (YF11700102) of the Key Laboratory for Water Environment and Resources, Tianjin Normal University.

References

1. S. L. Hardt, W. J. Rapaport, Recent and current artificial intelligence, in *J. the AI Magazine Summer*, (1986) pp. 91-100.
2. Z. Wang, Y. Liu, Design of road tracing navigation control for smart car use CCD sensor, *E-Health Networking*, in *Proc.2010 Int. Conf. Digital Ecosystems and Technologies*, (2010)pp. 345-348.
3. M. Tatsuya, H. Eiji, Navigation system for a mobile robot using an omni-directional camera, in *Proc. 12th Int. Symp. Artificial Life and Robotics*, (2007)pp. 450-453.
4. Y. Li, C. Yu, F. Dai, et al, Implementation of a versatile intelligent vehicle with tracking and remote control for hospital or nursing homes, in *Proc. 3rd IEEE Int. Conf. of Software Engineering and Service Sciences*, (2012)pp. 505-508.
5. MC9S12XS256 Reference Manual, Rev. 1.13 (2012), Freescale Semiconductor Inc..
6. J. J. Labrosse, *MicroC/OS-II: The real-time kernel*, 2nd edn, CMP Books(2003)