IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 8, August 2017

Study of µCOS RTOS on Low Cost Microcontroller

Pundaraja[1], Lavanya K.J[1], Priyanka D.L[1], Priyanka B.V[1]

M. Tech Student, Telecommunication Engineering (DCN), Dr. Ambedkar Institute of Technology, Bengaluru, India[1]

Abstract: Numbers of RTOS products from various vendors are available in the market, µC/OS-II a freeware with minimum facility is more popular among the hobbyist, researchers and small embedded system developers. In this project LCD is interfaced with the microcontroller using Serial interface. Four tasks are created using µC/OS-II RTOS. Task1 is for manage all other tasks from task create to delete. Task2 for reading characters from key board.Task3 for Displaying the entered characters on Hyper Terminal when ENTER key is pressed. Task4 for Displaying message on LCD when SPACE key is pressed. The software tools SDCC Compiler and ATMEL FLIP tool are used for implementation of tasks using µC/OS-II RTOS on AT89C51ED2 embedded development board. The result obtained indicates that best suitable for small scale industrial automation.

Keywords: SDCC, Keil IDE, µC/OS-II RTOS, Flash Magic and Embedded Development Board.

I. INTRODUCTION

The human effort needed for implementing µC/OS-II RTOS is less compared to other RTOS products. Among the available RTOS µC/OS-II RTOS is suitable for small scale embedded system using various controllers and processors, since it is a freeware and easily available. A micro-kernel operating system (µC/OS-II RTOS) written by John J Lebrosse is a freeware for peaceful research purpose. The source code is written in C language, which is compliant to ANSI 'C' format. It has about 10,000 lines of source code with well-documented comments in the source code. It is a well-tested source code. The µC/OS-II RTOS supports pre-emptive scheduling and not efficient with respect to processor utilization.

The scaled-version of µC/OS-II RTOS on 8051 with multiple tasks uses 4 KB of flash and 512 bytes of RAM. The based on the idea of only placing essential core real-time operating system functions in the kernel is of micro kernal and other functionality is designed in modules that communicate with the kernel via minimal defined interfaces. The results in easy reconfigurable systems without the need to rebuild the kernel in micro kernal. The µC/OS-II kernel is completely designed in software and is a well-used RTOS for combined set of software and hardware systems. It is described in a monolithic microkernel in the language of ANSI C with a small part in assembly language for context switching.

The objective of this project is to port µC/OS-II RTOS to 8051 controller and study the performance of multiple tasks. The performance of the 8051 microcontroller respect to different tasks is simulated in software platform and verified in the target hardware. To meet the above objectives, the software tools used for 8051 controller in this work are SDCC (Small device C Compiler) full version and Atmel Flip2.4.2. The µC/OS-II RTOS is loaded to the system and different tasks are executed.

II. PROPOSED WORK

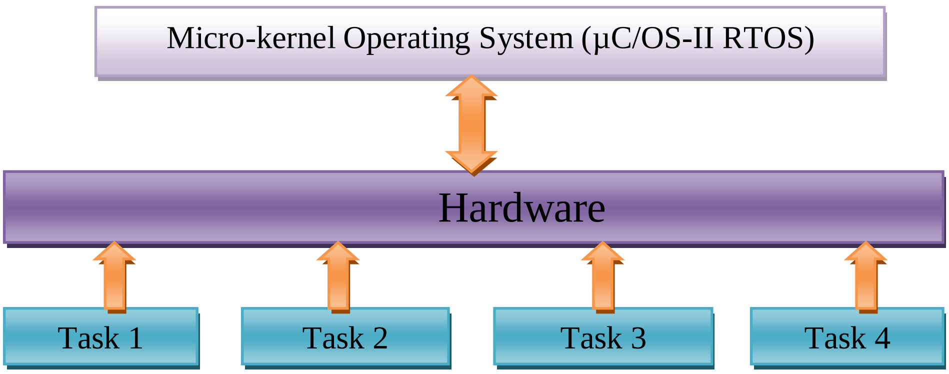

Fig. 1: Block diagram of handling of several tasks

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 8, August 2017

The proposed work shown in figure 1 is implemented on a low-end hardware. The figure 1 shows conceptual view of a scheduler which has to handle several tasks. The role of the scheduler is to handle several tasks by meeting all the deadlines. Several types of tasks like reading a key from a keypad, displaying messages on LCD, displaying messages on Hyper terminal. µC/OS-II RTOS is implemented on 8051 board.
The SDCC compiler is an improved version of C Compiler which can generates the Hex file with less memory as compared to other compilers. The advantage of using this is thus generated hex file can be downloaded on to microcontroller hence we will get free memory to install µCOS-II RTOS on microcontroller.

III. µC/OS-II RTOS

One of the popular RTOS for the embedded system development is µC/OS-II For non-commercial use, µC/OS-II RTOS is also a freeware. Jean J. Labrosse designed it in 1992 and nowadays µC/OS-II is well developed for a number of applications. It is available from Micrium (www.micrium.com). The µC/OS-II name is obtained from MCOS. It is also popularly known as µC/OS-II or MicroCOS or UCOS. Micrium describes portable µC/OS-II, it's able to work as ROM, scalablity, preemptivability, with real-time and multitasking kernel. µC/OS-II has been used in over thousands of applications, including automotive, avionics, electronics, the devices which are used in medical, military purpose, aircraft and aerospace, and networking, and systems-on-a-chip development. µC/OS-II has an elegant code and it gives good performance ratio with high quality. The USA defence department certified the code for widely use in medical applications and in avionics. It has a precertifiable software component for safety critical systems, including avionics system ADO-178B and EUROCAE ED-12B, medical FDA 510(k) for transportation and IEC 61058 standards is mainly for nuclear systems.Using this RTOS has another advantage. It has full source code availability and has been elegantly and very well documented in the book by its designer. Its code ports on many processors that are commonly used in the designing of embedded systems. µC/OS-II is real-time kernel with additional support as follows.
1. µC hardware peripherals.
2. µC/FL (an embedded flash memory loader).
3. µC/FS (an embedded memory file system).
4. µC/GUI (an embedded GUI platform).
5. µC/Probe (monitoring tool).
6. µC/TCPIP (TCP-IP stack).
7. µC/CAN (an embedded controller area network bus).
8. µC/MOD (an embedded modbus).
9. µC/USB device and µC/USB host (embedded USB-devices framework).

The Program is written in ANSI 'C' and is verified by the author that it can be compiled in most of the compilers. It is a freeware for peaceful research purposes only. For commercial purposes, royalty has to be paid to the author. It is a scalable operating system. It supports pre-emptive scheduler and lacks to support other schedulers. In this work, scheduler in µC/OS-II RTOS is used. µC/OS-II RTOS has 9,900 lines length of code. In this work, it is scaled to about 6,000 lines of code for 8051 controller. The scaling of operating system is performed by configuring the parameters in config.h file. The functions of the RTOS source code which can be configured by a user.

IV. PROCESSOR INDEPENDENT CODE

The processor independent code is the code which is applicable to any processor without any modifications. In µc/os-ii, this code is written in files as
1. OS_CORE.C
2. OS_MBOX.C
3. OS_MEM.C
4. OS_Q.C
5. OS_SEM.C
6. OS_TASK.C
7. OS_TIME.C
8. µC_II.C
9. µC_II.H

µC/OS-II header (included in master) and C files are µC_II.C and µC_II.H respectively. The files for the RTOS core, timer and task files are OS_CORE.C, OS_TIME.C and OS_TASK.C respectively. The files for the memory-partitioning, semaphore, queue and mailbox codes are OS_MEM.C, OS_SEM.C, OS_Q.C and OS_MBOX.C respectively.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified

Vol. 6, Issue 8, August 2017

V. HARDWARE SETUP OF 8051 BOARD

The hardware board as shown in figure 2 consists of functional blocks such as

1. Power supply
2. 8 LEDs
3. 4x4 Keypad
4. RS-232 port interface
5. 7 segment LED Display
6. 2x16 LCD

The input to power supply is given from a DC adapter of 5 volts. The rectified output of power supply block is given to microcontroller and other peripherals attached to it. LEDs can be programmed by setting the bit zero or one in any of the 8051 I/O ports to turn it ON/OFF. A 2x16 LCD consists of two rows. A keypad consists of 16 keys. A serial port interface is used to communicate between PC and target hardware. This is a serial communication interface and 9600 bps baud rate is used. Figure 2 shows the complete hardware setup used in this work. The 8051 microcontroller and PC interacts with each other using RS-232 port interface. In PC, the Windows operating system is used along with SDCC and Atmel Flip.

VI. RESULTS

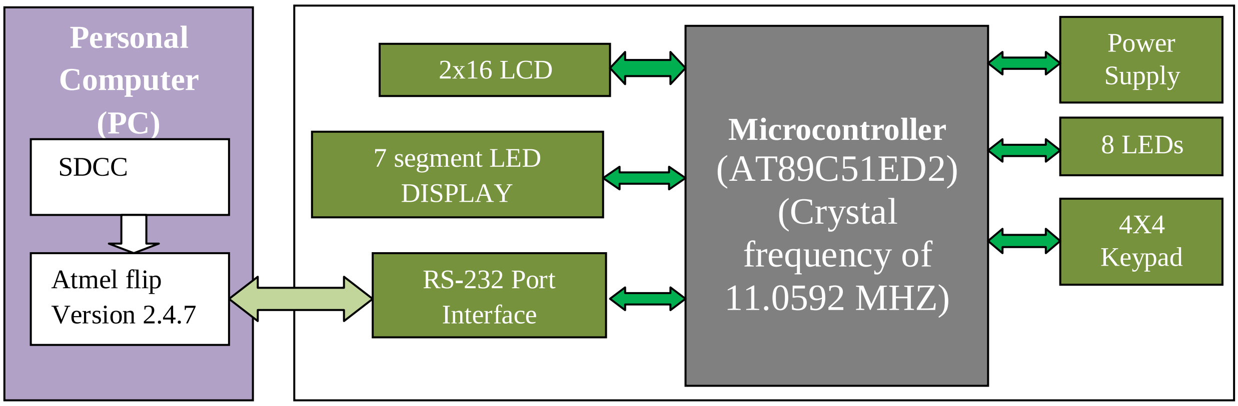

Fig. 2: Complete hardware setup of 8051 board

The program is compiled and loaded in to the AT89C51ED2 controller board. The corresponding results from the hardware are analyzed using theoretical approach and respective captured outputs are as shown below.

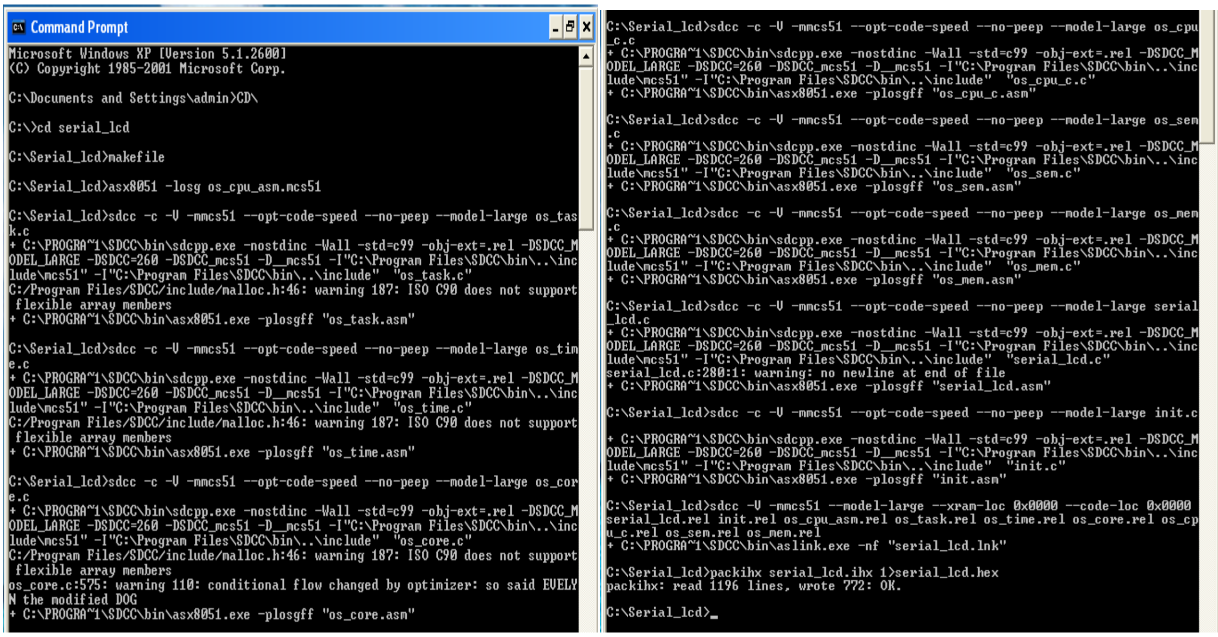

Fig. 3: Screen shot of project compilation in SDCC

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 8, August 2017

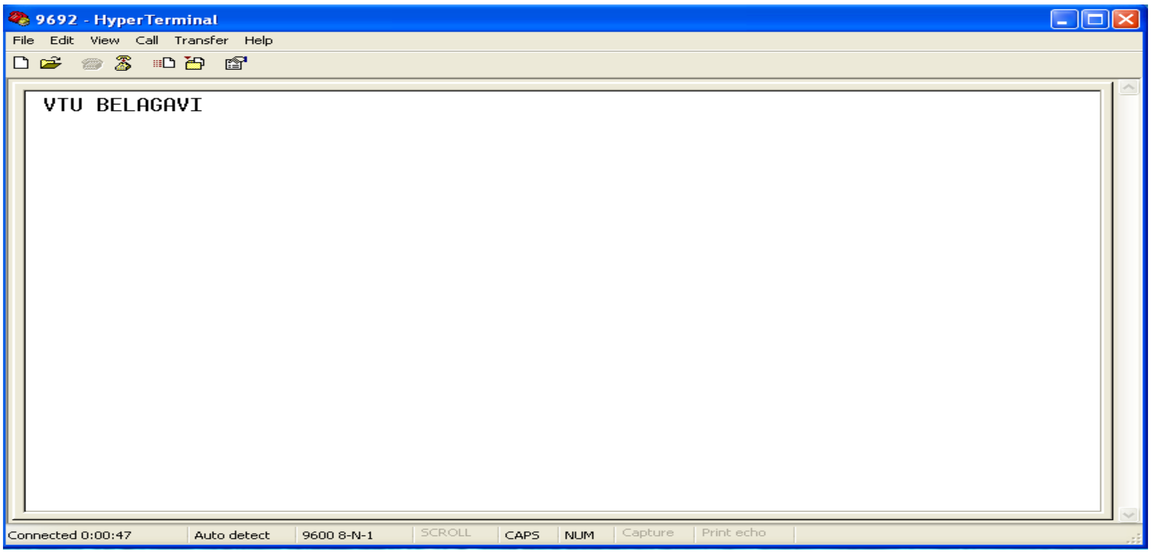

Fig. 4: Screen shot of output on HyperTerminal

Fig. 5: Screen shot of output on LCD display

VII. CONCLUSION AND FUTURE WORK

The SDCC compiler is advanced version of C Compiler which can generates the Hex file with less memory as compared to other compilers. The benefits of using this is thus generated hex file can be downloaded on to microcontroller hence we will get free memory to install µCOS-II RTOS on microcontroller. The results also show the ability to quickly and easily implement various other tasks by reconfiguring the microcontroller OS-II header file through the µC/OS-II RTOS.

The above work can be extended by designing a embedded system with GUI for the following

1. Toys with remote
2. Automated security system in banks to handle ATM transactions.
3. Photo-framer application
4. Encryptor/Decryptor to handle IP packets

REFERENCES

[1] R.R. Maggavi, D.A. Torse, "RM Analysis on ucos for Embedded Systems", International Journal of Latest Trends in Computing, volume 2, issue 1, March 2011.

[2] Albers, Susanne (1999). "Better bounds for online scheduling". SIAM Journal of Computing 29 (2): 459–473. doi:10.1137/S0097539797324874.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 8, August 2017

[3] S.M. Johnson, "Optimal two- and three-stage production schedules with setup times included", Naval Res. Log. Quart. I(1954)61-68.
[4] F. Engel, G. Heiser, I. KuZ, S. M. Petters and S. Ruocco, "Operating Systems on SoCs: A Good Idea?," in ERTSI in conjunction with 25th IEEE RTSS04, Lisbon, Portugal, December 2004.
[5] Furunas, J. "Benchmarking of a Real-Time System that utilises a booster." In International Conference on Parallel and Distributed Processing Techniques and Applications(PDPTA200),June,2000.
[6] L. Lindh, T. Klevin, and J. Furunas, "Scalable Architecture for Real- Time Applications – SARA". Swedish National Real-Time Conference SNART99 Linkoping, Sweden, August, 1999.

BIOGRAPHIES

Pundaraja was born on 8th October 1994, in Gulbarga, Karnataka, India. He studied his bachelor degree in Electronics and Communication Engineering from Maratha Mandal Engineering College, Belagavi, Karnataka, India in 2016 and currently pursuing his masters in Digital Communication and Networking in Dr.Ambedkar Institute of Technology, Bengaluru, Karnataka, India. His research interests include digital communication systems, Embedded systems design and Networking field.

Priyanka B.V was born on 21st January 1995, in Bengaluru, Karnataka, India. She studied her bachelor degree in Telecommunication Engineering from BMS College of Engineering, Bengaluru, Karnataka, India in 2016 and currently pursuing her masters in Digital Communication and Networking in Dr.Ambedkar Institute of Technology, Bengaluru, Karnataka, India. Her research interests include Microwaves and Antenna design and communication field.

Priyanka D.L was born on 24th May 1993, in Hassan, Karnataka, India. She studied her bachelor degree in Electronics and Communication Engineering from Rajeev Institute of Technology, Hassan, Karnataka, India in 2016 and currently pursuing her masters in Digital Communication and Networking in Dr.Ambedkar Institute of Technology, Bengaluru, Karnataka, India. Her research interests include Communication and Networking field.

Lavanya K.J was born on 4th October 1993, in Tumkur, Karnataka, India. She studied her bachelor degree in Electronics and Communication Engineering from Don Bosco Institute of Technology, Bengaluru, Karnataka, India in 2015 and currently pursuing her masters in Digital Communication and Networking in Dr.Ambedkar Institute of Technology, Bengaluru, Karnataka, India. Her research interests include Digital Communication and Networking field.