www.ijiarec.com

ISSN:2348-2079

Volume-6 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

Raspberry PI GUI kernel remote authentication multitasking & embedded web control

#1B.Guga Priya,*2Mr. B. PRABHAKARAN
PG Scholars, * professor
Department of Electrical and Electronics Engineering
Nandha Engineering College (Autonomous), Erode.
[1]E-Mail: gugapriya92@gmail.com
[2]E-Mail: s.prabhakaran2006@gmail.com

ABSTRACT- The system provide secured data transfer by remote method & each user can split the task an access and run the task by using OS which include IOT enable remote. The main features of the system includes Raspberry Pi Kernal Tasking & Control Network Remote Authentication Web Control By Master & Remote System. The software output is done with ARDUINO, ESP CORER 1.8 using embedded C. schematic is prepared using PROTEUS. Also the Software and hardware portion is done.

KEYWORDS—IOT,RASSPBERRY PI3, ARM 7, ATMEGA, VIDEO CORE IV,BCM 548 ,ESP8266-II,BUCK BOOST CONVERTER, HDMI TO VGA CONVERTER, RELAY DRIVER,GUI POWER AMPLIFIER , OPTO COUPLER, ETHERNET, LCD, PROTEUS, PYTHON IDE,ARDUNIO IDE, WIRE SHARK.

I. INTRODUCTION

The Internet of things (IoT) is the network of physical devices, vehicles, and other items embedded with electronics, software, sensors, actuators, and network connectivity which enable these objects to collect and exchange data. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. Experts estimate that the IoT will consist of about 30 billion objects by 2020. The IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. When IoT is augmented with sensors and actuators, the technology becomes an instance of the

more general class of cyber-physical systems, which also encompasses technologies such as smart grids, virtual power plants, smart homes, intelligent transportation and smart cities.

KERNAL MULTITASKING & CONTROL

Kernal is an operating system, is allowing a user to perform more than one computer task (such as the operation of an application program) at a time. The operating system is able to keep track of where you are

in these tasks and go from one to the other without losing information.

NETWORK REMOTE AUTHENTICATION

Network Level Authentication is a technology used in Remote Desktop Services (RDP Server) or Remote Desktop Connection (RDP Client) that requires the connecting user to authenticate themselves before a session is established with the server.

WEB CONTROL BY MASTER & REMOTE SYSTEM

It means control of a machine or apparatus from a distance by means of radio or infrared signals transmitted from a device.

II EXISTING SYSTEM

The main objective of the system,

To implement a pure kernel system in an Empty manner for creates an efficient platform for device driver.To make and configure they image data and beagle bone setup in terminal window.Unless the hardware being controlled is really misbehaving.User memory is swappable, unlike kernel memory. An infrequently used device with a huge driver won't occupy RAM that other programs could be using, except when it is actually in use. A well-designed driver program can still allow concurrent access to a device.

A.Drawbacks

- One user can access the data
- More drivers are required to execute the multitasking.
- Limited Number of users can access the Wi-Fi

B.Algorithm for Empty Kernel

In Linux operating system will able to execute the instructions in the terminal window. Here various parameter and command sets will run in the terminal window. Creating a directory setup updating the essential packages. Then install Yocto project simulator tool is prospective manner from the company website. Formation of an image in the minimal execution method. Device driver bridge module will be created by run qemux module. More executed parameter run on the executed terminal window.

1) STEP 1 - go to terminal and connect to internet
2) STEP 2 - sudo apt-get update
3) STEP 3 - sudo apt-get install build-essential
4) STEP 4 - Git clone -b dylan git://git.yoctoproject.org/pokyt
5) STEP 5 - cd poky (getting into the folder of yocto)
6) STEP 6 - source oe-init-build-env build-tamil-arm-simulation (creating a build directory in the name of yours)
7) STEP 7 - cd build-tamil-arm-simulation (getting into that directory)
8) STEP 8 - cd conf (getting into conf directory)
9) STEP 9 - geditlocal.conf (editing the file)
10) STEP 10 - change the line following
11) STEP 11 MACHINE ??= "qemux86" to MACHINE ?= "qemuarm"
12) STEP 12 - save and quit gedit
13) STEP 13 - cd.. (comming out of the conf directory)
14) STEP 14 - bitbake -k core-image-minimal (compiling ---- it will take more time to download and compile)

15) STEP 15- runqemuqemuarm (running the simulation)

III PROPOSED SYSTEM

The main objective of the system,

- Real time kernel based multitasking is performed in proposed system.
- Internet of things are used to control the parameter operations from any where.
- Remote authentication is used for the multiprocessor operation.

BLOCK DIAGRAM

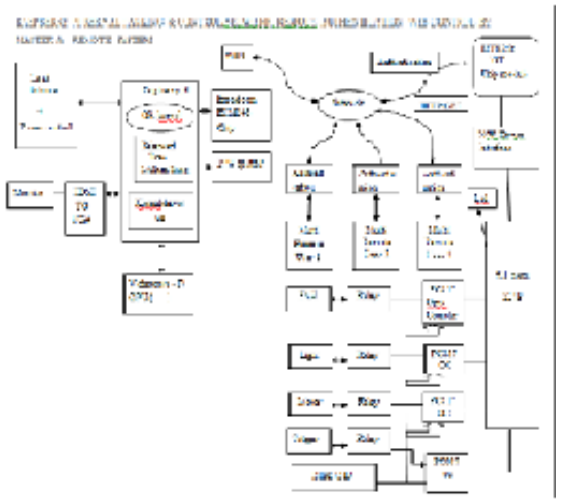


Figure 3.1 Block diagram of kernel multitasking

A.RASPBERRY PI 3

Figure 3.2 RASPBERRY PI 3

- Broadcom BCM2837 64 bit ARMv7 Quadcore processor powered single board computer running at 1.2 GHz.
- 1 GB RAM,Blutooth Low energy on board.
- 40 pin extented GPIO.
- Full size HDMI.
- CSI camera port for connecting the rassberry pi camera .
- DSI display port for connecting the rassberry pi touch screen display Micro SD card for loading your OS and storing data.

B.ATMEGA

Figure 3.3 ATMEGA

The Atmel 8-bit AVR RISC-based microcontroller combines 32 kB ISP flash memory.It contains read-while-write capabilities, 1 kB EEPROM, 2 kB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters.It contains compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter programmable watchdog timer with internal oscillator, and five software selectable power saving modes.The device operates between 1.8-5.5 volts. The device achieves throughput approaching 1 MIPS per MHz.

C. ESP 8266

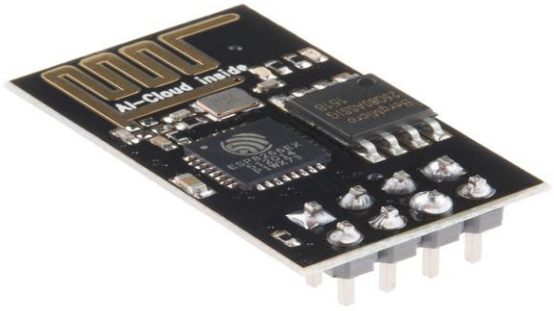


Figure 3.4 ESP8266

- The ESP8266 is a low-cost Wi-Fi microchip withfull TCP/IP stack and microcontroller capability
- Standards FCC/CE/TELEC/SRRC
- Protocols 802.11 b/g/n/e/i
- Frequency Range 2.4G ～ 2.5G (2400M ～ 2483.5M)
- Tx Power
- 802.11 b: +20 dBm
- 802.11 g: +17 dBm
- Rx Sensitivity
- 802.11 b: -91 dbm (11 Mbps)
- 802.11 g: -75 dbm (54 Mbps)

D. ARDUINO UNO R3

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License(GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself kits. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

Figure 3.5 Arduino Uno R3 Board

Arduino Uno R3 is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.. You can tinker with your UNO without worring too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again. "Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

E. LIQUID CRYSTAL DISPLAY

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data. The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a LCD.

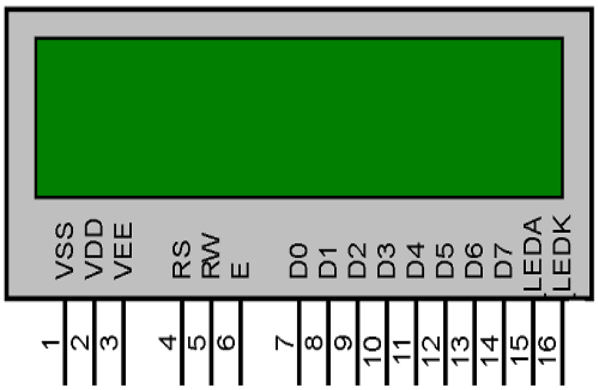


Figure 3.6 16x2 Lcd Display

IV TESTING

A.PROTEUS CIRCUIT FOR MULTITASKING ANALYSIS

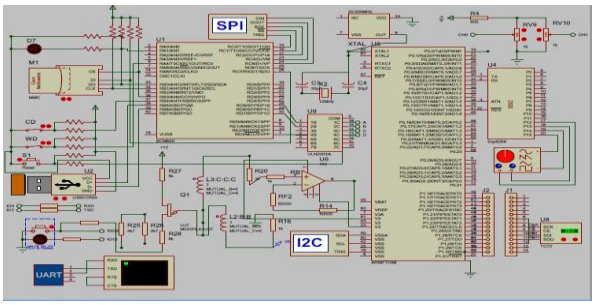


Figure 4.1 Multitasking analysis

B. PROTEUS CIRCUIT FOR RELAY

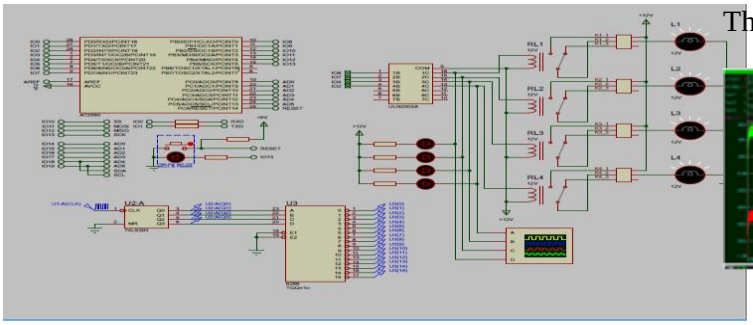


Figure 4.2 Relay circuit analysis

C.PROTEUS CIRCUIT FOR SECURED TRANSMISSION ANALYSIS

PARAMETER	EXISTING SYSTEM	PROPOSED SYSTEM
Remote user	Single user	Multi user
Kernal Size	2MB	1.5 MB
Threading	Single Thread	Multi Thread

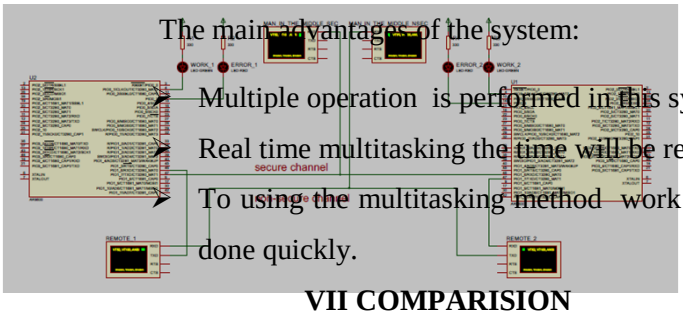
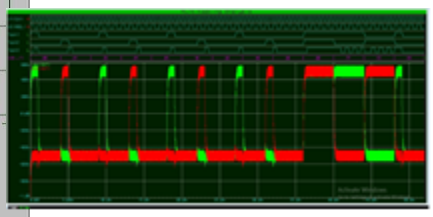


Figure 4.3 secured transmission analysis

V RESULT AND ANALYSIS

The simulation output is shown.



- Response time is 2 – 3 ms.
- A curve will be split in to 3 task process.

VI ADVANTAGES

The main advantages of the system:

- Multiple operation is performed in this system.
- Real time multitasking the time will be reduced.
- To using the multitasking method work will be done quickly.

VII COMPARISION

VIII CONCLUSION

Embedded Linux is an essential platform for advanced real world interfaces.Remote authentication is used for access the data and run the task quickly.Secured level data transmission is performed in this system.IOT enable remote user can access & control the such type of device in OS.Cost analysis done.

VIII FUTURE WORK.

The kernel Remote Authentication Multitasking can be further enhanced to real time application and implementation of software , hardware portion.

REFERENCES

1. K.Tamilselvan, M.E., (ph.d), "Real Time Kernel Based Hot Spot Communication Using Raspberry Pi" IJSRD Vol. 3, Issue 02, 2015 .

2. John Shawe Taylor, Theodoros Evgeniou,Charles A. Micchelli', "Learning Multiple Tasks with Kernel Methods", Journal of Machine Learning Research 6 (2005) 615–637 Submitted 2/05; Published 4/05.

3. Jean-Philippe, VertElsa Bernard,Yunlong Jiao','Kernel multitask regression for toxicogenetics", Aug 1,2017.

4. By Jie Liu , Edward A. Lee "Timed Multitasking for Real-Time Embedded Software" IEEE Control Systems Magazine , February 2003,10535888/03/2003.

5. Ralph Moore ,Mr. Moore, "How to Use Real-Time Multitasking Kernels in Embedded Systems" Micro Digital Associates,October 7, 2001.

6. AndiKleen "On submitting kernel patches" article 2010.

7. Andrew Morton kernel.org development and the embedded worldhttp://userweb.kernel.org/~akpm/rants/elc-08.odp.

8. Jonathan Corbet, "Linux Kernel Development" A White Paper By The Linux Foundation December 2010.

9. RajendraPrasad.M, S. Ramasubba Reddy, V.Sridhar. "Framework To Port Linux Kernel On power pc Based Embedded System Used OrtelecomApplication – Ipbts" International Journal of Software Engineering & Applications (IJSEA), Vol.2, No.4, October 2011.

10. Tamilselvan.K, G.Sivasangari and S.Menakambal "Multitasking Operating System for ARM Processors" with LED Display Vol. 3 issue.1 No.13, March 2014, 65-70.

11. http://www.ijisr.issr-journals.org.

12. http://www.sciencedirect.com/science/journals/all .