



SEDE GUAYAQUIL

FACULTAD DE INGENIERÍAS

CARRERA DE INGENIERÍA ELECTRÓNICA

**TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELECTRÓNICO**

TEMA:

**“DISEÑO DE APLICACIONES DE SISTEMAS EMBEBIDOS BASADOS EN
TECNOLOGIA RASPBERRY-PI Y ODROID-U3”.**

AUTORES:

**Martillo Ayala Daniel Humberto
Zambrano Mendoza Erly Loberty**

DIRECTOR:

ING. Luis Córdova Rivadeneira.

GUAYAQUIL, FEBRERO DEL 2015

DECLARATORIA DE RESPONSABILIDAD

Nosotros, Daniel Humberto Martillo Ayala portador de cédula de identidad No 0916315682 y Erly Loberty Zambrano Mendoza portador de cédula de ciudadanía 0916189541 estudiantes de la Universidad Politécnica Salesiana declaramos que la responsabilidad del contenido de este trabajo de titulación, corresponde exclusivamente a los autores y la propiedad intelectual es de la Universidad Politécnica Salesiana.

Guayaquil, Marzo del 2015

Daniel Humberto Martillo Ayala
CI: 0916315682

Erly Loberty Zambrano Mendoza
CI: 0916189541

DEDICATORIA

El presente trabajo está dedicado principalmente a mi Dios que día a día me da fuerza para continuar mejorando en todos los aspectos de mi vida.

A toda mi familia, mi esposa y mi hija pilar fundamental en cada meta que quiero alcanzar.

Mis padres que con amor y paciencia han forjado mi carácter y la persona de bien que ahora soy.

A mis verdaderas amistades que han estado siempre conmigo, apoyándome en los buenos y malos momentos de mi carrera de ingeniería.

No puedo dejar de lado a todos los docentes de la Universidad Politécnica Salesiana, que en cada clase o materia dictada, contribuyeron con mi desarrollo profesional y motivaron mi anhelo a aprender y descubrir cosas nuevas en este largo camino de estudios.

Erly Zambrano

DEDICATORIA

Este trabajo de grado y todo el esfuerzo realizado en él, está dedicado en primer lugar a DIOS por darme la fuerza y la fe necesaria para superar los diversos obstáculos que diariamente se presentan en la vida.

A mis padres que con mucho esfuerzo y amor me han guiado y aconsejado para lograr alcanzar mis metas y objetivos de bienestar.

A toda mi familia, por el infinito e incondicional apoyo y respaldo que me brindan a diario.

A mis amigos y compañeros que han compartido duros momentos en la formación profesional y académica.

Daniel Martillo Ayala.

AGRADECIMIENTO

El agradecimiento ponderado también de este trabajo para mi Dios, que me ha llenado de entendimiento para comprender cada cosa en mi vida y poder transformarlo en sabiduría.

A mi esposa Mabel Monserrate y a mi hijita Allison Zambrano, ellas que se han convertido en mi norte, en mi camino de mejora a seguir, agradezco todo el tiempo y tesón dedicado por ellas y el sacrificio del tiempo que han sufrido por dejarme alcanzar esta meta profesional.

A mi madre Cielo Mendoza, pues sé que con su lucha, dedicación y esfuerzos en sus hijos, en mí, no podría ahora escribir ninguna de estas líneas que ahora se plasman en este papel.

A mi compañero de tesis, que ha dejado su gran aporte en este trabajo en equipo para poder sacar adelante este meticulado y ambicioso tema de tesis.

Al profesor tutor de este presente trabajo de Tesis el Ing. Luis Córdova, que ha podido mayormente poner nuestra cabeza en frío en cada complicación que tuvimos en el camino previo a la concertación de este trabajo de Tesis.

Gracias

Erly Zambrano

AGRADECIMIENTO

En primer lugar quiero agradecer a DIOS por la oportunidad y bendición que he tenido en la vida de poder culminar mis estudios superiores y aplicarlos para el mejor bienestar y desarrollo de mi patria.

A mi madre Teresa Ayala y a mi padre Nilo Martillo, por su apoyo y motivación constante y por todas las preocupaciones y molestias que se tomaron para que pudiera alcanzar mi meta profesional.

A mis hermanas, porque me demostraron que una familia unida supera todos los obstáculos por muy difíciles que se presenten.

A mi compañero de tesis, por la confianza depositada y el respaldo fraterno brindado, por medio del cual se logró que este trabajo en equipo pueda superar todos los obstáculos presentados y alcanzar el objetivo final propuesto.

Al profesor tutor de este presente trabajo de tesis el Ing. Luis Córdova, que de manera comprometida brindó todo el respaldo, motivación y apoyo necesarios para alcanzar nuestro título profesional.

Gracias

Daniel Martillo Ayala.

ÍNDICE GENERAL

CARÁTULA	I
DECLARATORIA DE RESPONSABILIDAD	II
DEDICATORIA	III
AGRADECIMIENTO	V
ÍNDICE GENERAL.....	VII
ÍNDICE DE TABLAS	XIV
ÍNDICE DE FIGURAS.....	XV
RESUMEN.....	XXI
ABSTRACT.....	XXIII
INTRODUCCIÓN	1
CAPÍTULO 1	2
EL PROBLEMA	2
1.1 Planteamiento del Problema.....	2
1.2 Delimitación del Problema	2
1.3 Objetivos	3
1.3.1 Objetivo General	3
1.3.2 Objetivos Específicos.....	3
1.4 Justificación.....	4
1.5 Hipótesis.....	5
1.6 Variables e Indicadores	5
1.6.1 Variable Dependiente.....	6
1.6.2 Variable Independiente.	6
1.7 Metodología	6
1.7.1 Métodos.....	6

1.7.1.1 Método Experimental.....	6
1.7.1.2 Método Deductivo.....	6
1.7.2 Técnicas.....	6
1.7.2.1 Técnica Documental.....	7
1.7.2.2 Técnica de Campo.....	7
1.7.3 Instrumentos de Investigación y recolección de datos	7
1.8 Población y Muestra.....	7
1.9 Descripción de la propuesta	7
1.10 Beneficiarios	8
1.11 Impacto.....	9
CAPÍTULO 2	10
MARCO TEÓRICO.....	10
2.1 HARDWARE.....	10
2.1.1 ¿Qué es Rapberry-pi?.....	10
2.1.2 Tipos de Raspberry Pi	10
2.1.2.1 Raspberry Pi Modelo A.....	11
2.1.2.2 Raspberry Pi Modelo A+	12
2.1.2.3 Raspberry Pi Modelo B	14
2.1.2.4 Raspberry Pi Modelo B +.....	15
2.1.2.5 Raspberry-Pi 2 Modelo B	18
2.1.3 De qué está hecha la Raspberry Pi tipo B	20
2.1.3.1 Tarjeta de red.....	20
2.1.3.2 Conexiones.....	20
2.1.4 Especificaciones Raspberry Pi tipo B	21
2.1.5 ARQUITECTURA.....	22
2.1.5.1 ARM vs. x86	22

2.1.5.2 Windows vs. Linux	24
2.1.6 Software para Raspberry Pi.....	25
2.1.7 ACCESORIOS	29
2.1.8 ¿Qué es ODROID?.....	30
2.1.8.1 Tipos de Odroid	31
2.1.8.1.1 ODROID-U3	31
2.1.8.1.2 ODROID-XU3	32
2.1.8.1.3 ODROID-C1	33
2.1.9 ¿ De qué está hecha la ODROID U3 ?.....	34
2.1.9.1 Características principales.....	34
2.1.9.2 Conociendo la Placa.....	35
2.1.9.3 Los Periféricos	37
2.1.10 ¿Por qué ARM?.....	41
2.1.11 ¿Por qué ODROID?	42
2.1.12 Tecnología big.LITTLE	42
2.1.13 Problemas ODROID	44
2.1.13.1 Características Técnicas	44
2.1.14 ODROID-U3 VS RASPBERRY PI.....	46
2.1.14.1 Hardware	46
2.1.14.2 Rendimiento	48
2.1.14.3 Programación y Desarrollo	49
2.1.14.4 Conclusión	50
2.2 SOFTWARE	51
2.2.2 ¿Qué es Qt Creator?	51
2.2.1.1 ¿Por qué QT?.....	52
2.2.1.2 Características fundamentales de Qt Creator:	52

2.2.1.3 ¿Cómo está disponible Qt?.....	54
2.2.1.4 ¿Cómo obtener e instalarse Qt?.....	55
2.2.2 ¿Qué es Python?.....	55
2.2.2.1 ¿Por qué Python?	56
2.2.2.2 Características fundamentales de Python:.....	56
2.2.2.3 Instalación de Python	58
2.2.3 ¿Qué es Scilab?	59
2.2.3.1 Xcos	61
2.2.3.2 Paquetes disponibles para Scilab (Toolbox)	62
2.2.3.3 Instalación de Scilab.	63
2.2.4 ¿Qué es GNU OCTAVE?	64
2.2.4.1 Detalles técnicos:	66
2.2.4.2 Instalación en GNU/Linux	68
CAPÍTULO 3	69
ESTRUCTURA DEL DESARROLLO DE LAS PRÁCTICAS.	69
3.1 Estructura de los kits para el desarrollo de las prácticas	69
3.2 Componentes que lo conforman.....	69
CAPÍTULO 4	72
PRÁCTICAS Y RESULTADOS DEL PROYECTO	72
PRÁCTICA No 1	73
CONFIGURACIÓN INICIAL E INSTALACIÓN DE PROGRAMAS EMPLEADOS EN CONTROL ELECTRÓNICO.....	73
1. OBJETIVO GENERAL	73
2. OBJETIVOS ESPECÍFICOS.....	74
3. MARCO TEÓRICO.....	74
4. DESARROLLO	76
5. CONCLUSIONES.	107

PRÁCTICA No 2	108
CONFIGURACIÓN DE LA RED INALÁMBRICA Y SISTEMAS DE VÍDEO VIGILANCIA.	108
1. OBJETIVO GENERAL.	109
2. OBJETIVOS ESPECÍFICOS.	109
3. MARCO TEÓRICO.	109
4. DESARROLLO.	110
5. CONCLUSIONES.	119
PRÁCTICA No 3	120
CONFIGURACIÓN Y USO DE LOS PINES DIGITALES DEL PUERTO GPIO.	120
1. OBJETIVO GENERAL.	120
2. OBJETIVOS ESPECÍFICOS.	121
3. MARCO TEÓRICO.	121
4. DESARROLLO.	125
5. CONCLUSIONES.	134
PRÁCTICA No 4	135
CONFIGURACIÓN Y USO DE LOS PINES UARTS PARA LA COMUNICACIÓN RS-232 EN EL PUERTO GPIO.	135
1. OBJETIVO GENERAL.	135
2. OBJETIVOS ESPECÍFICOS.	136
3. MARCO TEÓRICO.	136
4. DESARROLLO.	139
5. CONCLUSIONES.	142
PRÁCTICA No 5	142
INTRODUCCIÓN A PHYTON	143
1. OBJETIVO GENERAL	143
2. OBJETIVOS ESPECÍFICOS.	143

3. MARCO TEÓRICO	143
4. DESARROLLO	145
5. CONCLUSIONES.	151
PRÁCTICA No 6	152
CONTROL DE UN MOTOR DE PASOS Y DE UN SERVOMOTOR PROGRAMANDO EN PYTHON.....	152
1. OBJETIVO GENERAL.	152
2. OBJETIVOS ESPECÍFICOS.....	152
3. MARCO TEÓRICO.....	153
4. DESARROLLO.	154
5. CONCLUSIONES.	158
PRÁCTICA No 7	159
INTRODUCCIÓN A GNU Octave y Scilab/Xcos	159
1. OBJETIVO GENERAL.	160
2. OBJETIVOS ESPECÍFICOS.....	160
3. MARCO TEÓRICO.....	160
4. DESARROLLO.	163
5. CONCLUSIONES	181
PRÁCTICA No 8	182
INTRODUCCIÓN A QT CREATOR Y DESARROLLO DE INTERFACES GRÁFICAS EMPLEADAS EN INSTRUMENTACIÓN VIRTUAL Y COMPARACIÓN CON LAB-VIEW.....	182
1. OBJETIVO GENERAL	183
2. OBJETIVOS ESPECÍFICOS.....	183
3. MARCO TEÓRICO.....	183
4. DESARROLLO.	187
5. CONCLUSIONES.	199
CONCLUSIONES	201

RECOMENDACIONES	203
CRONOGRAMA	204
PRESUPUESTO	205
BIBLIOGRAFÍA	206
ANEXOS	208
GLOSARIO	209

ÍNDICE DE TABLAS

Tabla 2.1: Especificaciones raspberry pi	21
Tabla 2.2: Lista de software que pueden ser instalados en su embebido raspberry... ..	25
Tabla 2.3 Tabla de características	44

ÍNDICE DE FIGURAS

Figura 2.1 Raspberry Pi Modelo A	12
Figura 2.2 Raspberry Pi Modelo A+	13
Figura 2.3 Diferencias entre modelos	13
Figura 2.4 Modelo B	14
Figura 2.5 Características & costos.....	15
Figura 2.6 Modelo B+	16
Figura 2.7 Diferencias Modelo B & Modelo B+	16
Figura 2.8 Raspberry Pi2 Modelo B	18
Figura 2.9 En perspectiva Raspberry Pi2 Modelo B.....	19
Figura 2.10 Accesorios que complementan la Raspberry	30
Figura 2.11 Odroid U3	32
Figura 2.12 Odroid XU3	32
Figura 2.13 Odroid C1	33
Figura 2.14 Partes placa Odroid U3.....	35
Figura 2.15 Fuente de alimentación Odroid U3.....	38
Figura 2.16 Micro SD Odroid U3	38
Figura 2.17 Modulo eMMC	38
Figura 2.18 Cable HDMI a micro HDMI.....	39
Figura 2.19 Adaptador Wifi	39
Figura 2.20 Case Odroid U3	40
Figura 2.21 Arquitectura Odroid U3.....	41
Figura 2.22 Case Odroid U3	47
Figura 2.23 Comparación entre ODROID & RASPBERRY.....	48
Figura 2.24 Comparación de Rendimiento	49

Figura 2.25 Comparación de Rendimiento	50
Figura 2.26 Entorno Qt Creator	51
Figura 2.27 Qt creator	53
Figura 2.28 Logo python.....	55
Figura 2.29 Programar Python	58
Figura 2.30 Log Scilab.....	60
Figura 2.31 Entorno Scilab	60
Figura 2.32 Funciones en Scilab	62
Figura 2.33 Navegando en Scilab	62
Figura 2.34 Módulos en Scilab	63
Figura 2.35 Instalación en Scilab.....	64
Figura 2.36 Logo GNU OCTAVE.....	65
Figura 2.37 Entorno GNU OCTAVE	67
Figura 3.1 Equipos necesario dentro de maleta contenedora.....	70
Figura 4.1 Logo de Equipos empleados.....	73
Figura 4.2 Fusión y resultado de Raspbian	74
Figura 4.3 Fusión y resultado de lubuntu.....	76
Figura 4.4 Sitio de descarga Raspberry	77
Figura 4.5 Archivo Imagen	77
Figura 4.6 Archivo extensión.....	78
Figura 4.7 Descarga Win32DiskImager.....	78
Figura 4.8 Descompresión Win32DiskImager.....	79
Figura 4.9 Abrir Win32DiskImager.....	79
Figura 4.10 Entorno Win32DiskImager.....	80
Figura 4.11 Ventana sobrescribir	80
Figura 4.12 Progreso de avance.	81

Figura 4.13 Indicación de terminación de Proceso	81
Figura 4.14 Opciones Menú Raspberry	83
Figura 4.15 Cambio de contraseña.....	84
Figura 4.16 Cambio al modo gráfico	85
Figura 4.17 Cambio al modo gráfico	85
Figura 4.18 Selección opción I1.....	86
Figura 4.19 Selección opción I2.....	87
Figura 4.20 Selección opción I3.....	88
Figura 4.21 Habilitación de la cámara	89
Figura 4.22 Rastrack	90
Figura 4.23 Selección Overclock.....	91
Figura 4.24 Menú Opciones Avanzadas	91
Figura 4.25 Opción A2 Hostname	92
Figura 4.25 Opción A3 Distribución de Memoria	93
Figura 4.26 Ingreso de “Login-Password”.....	94
Figura 4.27 Ejecución modo Gráfico.....	94
Figura 4.28 Sitio descarga Lubuntu.	96
Figura 4.29 Archivo extensión.....	96
Figura 4.30 Entorno Win32DiskImager.....	97
Figura 4.31 Confirmación de ruta y sobre escritura.....	97
Figura 4.32 Progreso de avance.	98
Figura 4.33 Indicación de terminación de Proceso	98
Figura 4.34 Entorno gráfico LXDE	99
Figura 4.35 Ingreso a Odrid Utility.....	100
Figura 4.36 Selección salida HDMI.....	100
Figura 4.37 Actualización Kernel	101

Figura 4.38 Configuración del idioma	102
Figura 4.39 Selección tipo de Teclado	102
Figura 4.40 Selección tipo de Teclado	103
Figura 4.41 Ruta para modificación de archivo	103
Figura 4.42 Modificación Archivo Locale	104
Figura 4.43 Comando cambio de fecha y hora	105
Figura 4.44 Selección de ubicación geográfica.....	105
Figura 4.45 Logo software Motion.	108
Figura 4.46 Fragmento URL	111
Figura 4.47 Uso de Ip estática.....	113
Figura 4.48 daemon on (por defecto está off)	116
Figura 4.49 webcam_localhost off (por defecto está on).....	117
Figura 4.50 locate on (por defecto está off)	117
Figura 4.51 width 640 y height 480	118
Figura 4.52 Cambio de Puerto.	119
Figura 4.53 Captura de imagen.	119
Figura 4.54 Logo de equipos empleados.	120
Figura 4.55 Pin out.....	121
Figura 4.56 Contenido directorio	123
Figura 4.57 Corrida de comandos	124
Figura 4.58 Revisión de contenido del directorio	124
Figura 4.59 Revisión de contenido del directorio	125
Figura 4.60 Activación desactivación de la salida.....	126
Figura 4.61 Armado de circuito de Prueba en el PROTOBOARD.....	126
Figura 4.62 Cambio de estado en el programa.....	127
Figura 4.63 Vista de la GPIO	128

Figura 4.64 GPIO montada con la ODROID U3	129
Figura 4.65 Diagrama de Bloque comunicación Odroid & GPIO	129
Figura 4.67 Diagrama de Conexión	133
Figura 4.68 Comunicación Serie.....	135
Figura 4.69 Conexión a los pines serie GPIO	137
Figura 4.70 Pines GPIO correspondientes a la comunicación serie.....	138
Figura 4.71 Comando para verificación de accesibilidad	139
Figura 4.72 Comando para verificación de proceso.....	140
Figura 4.73 Línea a comentar en fichero /etc/inittab.	141
Figura 4.74 Edición de fichero /boot/cmdline.txt	142
Figura 4.75 Logo de Equipos empleados.....	143
Figura 4.76 Aplicación de Ejemplo	144
Figura 4.77 Ejecución de ejemplo.....	145
Figura 4.78 Conectado en PROTOBOARD	146
Figura 4.79 Ejecución de los comandos.....	148
Figura 4.80 Acción guardar archivo	149
Figura 4.81 Ejecución de comando.....	149
Figura 4.82 Guardar archivo	150
Figura 4.83 Prueba final líneas de comando	150
Figura 4.84 Logo PYTHON.....	152
Figura 4.85 Motor paso a paso	154
Figura 4.86 Programas a utilizar	159
Figura 4.87 Captura de pantalla Octave	161
Figura 4.88 Pantallas de Scilab	163
Figura 4.89 Terminal de Octave.....	164
Figura 4.90 Ejecución de operaciones matemáticas Scilab.	170

Figura 4.91	Ventana de ayuda del programa	171
Figura 4.92	Lista de comandos más utilizados	172
Figura 4.93	Editor de Scripts	172
Figura 4.94	Integración del Editor en Scilab	173
Figura 4.95	Ingreso a Xcos	174
Figura 4.96	Ventana principal de Xcos.....	175
Figura 4.97	Explorador de Paletas	176
Figura 4.98	Ventana Simulación.....	177
Figura 4.99	Ventana de configuración de la simulación.....	177
Figura 4.100	Ejemplo de configuración del muestreo	179
Figura 4.101	Muestreo en ejecución	179
Figura 4.102	Bloques de integración	181
Figura 4.103	Logo programa QT CREATOR	182
Figura 4.104	Entorno QT Creator.....	185
Figura 4.105	Ventana principal de la Aplicación.....	188
Figura 4.106	Estructura en QT CREATOR	189
Figura 4.107	Main.cpp	192
Figura 4.108	Pantalla Principal QT CREATOR	193
Figura 4.109	Editor de Formularios	197

RESUMEN

AÑO	ALUMNO/S	DIRECTOR DE TESIS	TEMA DE TESIS
2015	DANIEL HUMBERTO MARTILLO AYALA ERLY LOBERTY ZAMBRANO MENDOZA	ING. LUIS CÓRDOVA RIVADENEIRA, MSC.	“DISEÑO DE APLICACIONES DE SISTEMAS EMBEBIDOS BASADOS EN TECNOLOGÍA RASPERRY-PI Y ODROID-U3”

El siguiente trabajo de titulación: “DISEÑO DE APLICACIONES DE SISTEMAS EMBEBIDOS BASADOS EN TECNOLOGÍA RASPERRY-PI Y ODROID-U3” trata sobre la implementación de kits y desarrollo de prácticas de laboratorio, empleando sistemas embebidos, en diversas materias referentes a la carrera de ingeniería electrónica. El punto central de este trabajo lo constituyen los computadores embebidos Raspberry-Pi y ODROID-U3 sistemas embebidos modernos que se programan bajo diversos lenguajes de libre distribución.

El objetivo principal que persigue este trabajo es proporcionar herramientas para el desarrollo de prácticas de laboratorio en materias como electrónica, instrumentación, automatismos, control, telecomunicaciones, domótica, etc. utilizando nuevas tecnologías orientadas a la programación.

El diseño de las aplicaciones se basa en la necesidad de disponer de un conjunto de elementos fácilmente compatibles y modulares en los cuales se pueda configurar bajo programación estructurada las funciones de control a ejecutar.

Los beneficiarios de este trabajo, alumnos de Ingeniería Electrónica, afianzan sus conocimientos en programación bajo software de distribución libre de licencias y

profundizan sus conocimientos sobre nuevas tecnologías para el desarrollo de proyectos.

PALABRAS CLAVES: Diseño de aplicaciones, Implementación de Kits, Computadores Embebidos, domótica, programación estructurada, software de distribución libre de licencias.

ABSTRACT

DATE	STUDENTS	DIRECTED BY THESIS	THESIS TITLE
2015	DANIEL HUMBERTO MARTILLO AYALA ERLY LOBERTY ZAMBRANO MENDOZA	ING. LUIS CÓRDOVA RIVADENEIRA, MSC.	APPLICATION DESIGN OF EMBEDDED SYSTEMS BASED ON TECHNOLOGY RASPBERRY-PI AND ODROID-U3

The next work: "APPLICATIONS DESIGN OF EMBEDDED SYSTEMS BASED ON RASPBERRY-PI AND ODROID-U3 TECHNOLOGY" discusses the implementation of development kits and laboratory practice using embedded systems in various subjects relating to electronic engineering. The focus of this work is embedded computers Raspberry-Pi and ODROID-U3 modern embedded systems that are programmed under various languages free distribution.

The main objective of this work is to provide tools for the development of laboratory practices in areas such as electronics, instrumentation, automation, control, telecommunications, domotic, etc. using new technology oriented programming. The application design was based on the need for a set of modular elements easily compatible and in which structured programming could be set under the control functions to be executed.

The beneficiaries of this work, Electronic Engineering students, entrench their knowledge in programming under free software licenses and deepen their knowledge of new technologies for the development of projects.

KEYWORDS: Application Design, Implementation Kits, Embedded Computers, home automation, structured programming, free software licenses.

INTRODUCCIÓN

El siguiente trabajo trata la elaboración y desarrollo de diversas aplicaciones de los sistemas embebidos más populares a la fecha actual, para la realización de prácticas de laboratorio que abarcan varios temas en las materias elementales más representativas para la formación del profesional en ingeniería electrónica.

Los Sistemas Embebidos son dispositivos electrónicos usados para controlar y operar equipos, máquinas, aparatos domésticos, equipos móviles, automóviles, instrumentos electrónicos y hasta plantas industriales.

Es de gran importancia para los estudiantes el manejo de sistemas y computadores embebidos que permitan con facilidad establecer firmemente las habilidades de programación estrechamente relacionadas con los conceptos y nociones de control.

Las placas Raspberry- Pi y ODROID-U3 son maravillas en miniatura, aguardan en su interior un importante poder de cómputo en un tamaño no más grande que el de una tarjeta de crédito. Son capaces de realizar cosas extraordinarias, que van desde el entretenimiento en el hogar hasta el control industrial.

No importa si el estudiante es principiante o si es un programador experto.

Es por este motivo que se utilizan estas placas, para implementar los Kits didácticos de laboratorio, en el desarrollo de aplicaciones con sistemas embebidos.

CAPÍTULO 1

EL PROBLEMA

1.1 Planteamiento del Problema

La Universidad Politécnica Salesiana (UPS), Sede Guayaquil, es una entidad de educación superior ubicada en las calles Chambers y 5 de junio con especializaciones en las ramas científico-técnicas.

La carrera de Ingeniería Electrónica es una de las más trascendentales de las ingenierías y dentro de su pensum curricular se destacan materias como sistemas digitales, sensores e instrumentación, control y automatización industrial, áreas en las que se requiere estar con permanente y constante actualización de las nuevas tecnologías aplicables a sus funciones.

En estas áreas, los estudiantes requieren tener a su disposición, instrumental para ejecución de prácticas que afiancen los conocimientos teóricos impartidos y además facilite el desarrollo de proyectos utilizando sistemas que empleen tecnología actual vigente y de libre desarrollo.

En vista de las tendencias de desarrollo que se presentan en varios países Europeos y de primer mundo, a la apertura del crecimiento en el uso de software y hardware libre de licencias, se ha decidido implementar un conjunto de Kits de laboratorio para el desarrollo de aplicaciones de sistemas embebidos, el cual es un aporte significativo en el fortalecimiento y uso de nuevas tecnologías de los estudiantes de la carrera.

1.2 Delimitación del Problema

Actualmente los Sistemas Embebidos se encuentran formando parte sustancial de la vida cotidiana y miles son producidos y vendidos cada año, en forma de equipos médicos, módulos electrónicos en automóviles, teléfonos celulares, sistemas de navegación aérea, etc. Cada vez que se mira alrededor es posible encontrar un sistema embebido. Por estas razones es sumamente importante conocerlos. Con el

presente trabajo de investigación y desarrollo se proporcionan kits de laboratorio para el desarrollo de aplicaciones con Sistemas Embebidos lo que constituye un fortalecimiento e incremento de las prestaciones didácticas de la carrera de Ingeniería Electrónica.

El desarrollo se cubre con diez unidades didácticas para uso y aplicaciones de los sistemas embebidos Raspberry-Pi y ODROID-U3. Adicionalmente se incluye la redacción de ocho prácticas configuradas para proporcionar un incremento gradual de los conocimientos en el uso de las unidades.

Las prácticas y aplicaciones se orientan a temas concernientes a instrumentación, automatismo, robótica, control y domótica. Pero está delimitado a la prestación, por parte de la Universidad, de una conexión a Internet libre de Proxy y dedicada de preferencia exclusivamente a esta función.

1.3 Objetivos

1.3.1 Objetivo General

Implementar kits de laboratorio para la ejecución de prácticas y aplicaciones en electrónica y control con sistemas embebidos utilizando los computadores embebidos Raspberry-PI y ODROID-U3.

1.3.2 Objetivos Específicos

- Desarrollar de manera práctica los conocimientos en las diversas materias relacionadas a la electrónica analógica, digital, comunicaciones, control e instrumentación adquiridos a lo largo de la carrera de ingeniería electrónica aplicando sistemas embebidos.
- Promover y aplicar técnicas de Programación en Python, QT, C++ y varios lenguajes de programación multiplataforma y de libre distribución.
- Fomentar el desarrollo de proyectos para fortalecer y mejorar el aprendizaje de los diferentes capítulos técnicos tratados en diversas materias.

- Elaborar una guía completa de prácticas didácticas en las cuales se desarrolle de manera gradual el manejo y aplicación de los módulos del laboratorio.
- Fomentar y fortalecer el uso de software libre de licencias en el desarrollo de sistemas de monitoreo e instrumentación virtual.
- Adquirir y transmitir datos a través de los puertos de los sistemas embebidos para desarrollar aplicaciones de control.
- Establecer comparaciones entre las prestaciones y alcances existentes entre los programas de análisis matemático Mat-Lab y Octave - Scilab.
- Establecer comparaciones entre las prestaciones y alcances existentes entre los programas Lab-View y QT Creator para aplicaciones de monitoreo e instrumentación industrial.
- Facilitar el desarrollo e implementación de proyectos de robótica y domótica.

1.4 Justificación

La tendencia actual de los sistemas electrónicos se relaciona directamente con el concepto de modularidad en el diseño, lo cual está dirigido a utilizar soluciones tecnológicas que aunque complejas en su funcionamiento interno, cada vez son menos difíciles de diseñar y adaptar a problemas específicos.

Uno de los aspectos que más ha colaborado en este efecto es la relación cada vez más estrecha y dependiente entre el hardware y el software de aplicación libre de licencias. Es por esto que los dispositivos embebidos se convierten en una herramienta de desarrollo muy importante para nuestro entorno, al permitir concebir sistemas electrónicos avanzados de una manera más eficiente en cuanto al diseño y las características finales del sistema.

Entre los sistemas embebidos didácticos que actualmente gozan de mayor popularidad se encuentran los computadores Raspberry-Pi y ODROID-U3.

Raspberry-Pi y ODROID-U3 son ordenadores completos de muy bajo coste, y sus posibilidades son impresionantes: estación de trabajo, servidor multimedia, servidor

web, controlador de otros dispositivos, domótica, aplicaciones de control, robótica, etc.

La formación del alumnado en competencia digital y automatización industrial, se beneficia enormemente si desde un primer momento conocen el funcionamiento interno de un ordenador a la vez que aprenden los fundamentos de programación en nuevos y actuales lenguajes de software libre y sin requerimientos de licencias para su uso.

Por su versatilidad, la Raspberry-PI y ODROID-U3 son herramientas muy poderosas no solo en las asignaturas exclusivamente tecnológicas y de control electrónico sino en todas las que pueden hacer uso del internet y de la inteligencia colectiva en la que se ha transformado la computación en la nube.

Si se desea alumnos que utilicen la tecnología creativamente y no como meros receptores de contenidos de hardware y software elaborados por otros, es necesario abordar iniciativas que les permitan usar los recursos tecnológicos como actores y creadores de tecnología.

1.5 Hipótesis

El desarrollo de prácticas utilizando los kits de laboratorio implementados para aplicaciones con Sistemas Embebidos, fortalece el aprendizaje de los estudiantes de la carrera de ingeniería electrónica, afianzando sus conocimientos de una manera ágil y amena, orientándolos hacia el uso de la programación estructurada utilizando lenguajes multiplataforma y de libre distribución.

1.6 Variables e Indicadores

1.6.1 Variable Dependiente.

Implementación de Kits de laboratorio utilizando los sistemas embebidos Raspberry-Pi y ODROID-U3, elaborar las prácticas correspondientes a su empleo.

1.6.2 Variable Independiente.

Actualización y fortalecimiento del uso de nuevas tecnologías y sistemas en los estudiantes de Ingeniería Electrónica de la UPS sede Guayaquil.

1.7 Metodología

1.7.1 Métodos

Para el desarrollo de este trabajo se utilizan los siguientes métodos de investigación y técnicas aplicadas.

1.7.1.1 Método Experimental

Se ejecuta este método, al desarrollar pruebas preliminares para la configuración de cada práctica con los sistemas embebidos propuestos.

1.7.1.2 Método Deductivo

Al recolectar datos y tabularlos se deducen y concluyen los patrones de configuración para cada una de las prácticas en el presente trabajo y se complementan paso a paso los conocimientos necesarios en los sistemas embebidos propuestos.

De esta manera se configura, aprende comandos básicos, y se desarrolla el uso de la Raspberry –PI y de la ODROID - U3.

1.7.2 Técnicas

1.7.2.1 Técnica Documental

Se utiliza la técnica documental al recolectar la información existente en fuentes secundarias tales como revistas especializadas, sitios web, manuales, datasheets, textos guías, etc. para poder sustentar con las prácticas la operación de los sistemas embebidos utilizados, además de incluir y demostrar en el desarrollo del marco teórico las bondades y alcance del trabajo propuesto.

1.7.2.2 Técnica de Campo

La técnica de campo se desarrolla mediante la ejecución de ocho prácticas dispuestas en orden de desarrollo progresivo debido a que la misma está ligada con la observación y comprobación de lo manifestado dentro de la hipótesis y la verificación de esta es la conclusión positiva del alcance total del presente trabajo.

1.7.3 Instrumentos de Investigación y recolección de datos

Los instrumentos empleados en la recolección de los diversos datos son:

- Levantamiento del estado actual del mercado de sistemas embebidos y dispositivos necesarios.
- Hacer pruebas de funcionamiento de los kits y ajustar los niveles de exactitud y confiabilidad.
- Análisis de las mejoras a realizar.
- Análisis de las prácticas a ejecutar como complemento y mejora de los
- Tópicos tratados en diversas asignaturas de la especialización electrónica.
- Búsqueda de los datos y características técnicas de los dispositivos a utilizar.

1.8 Población y Muestra

Este proyecto es de mucha utilidad para la población de estudiantes de la carrera de ingeniería electrónica de la universidad politécnica salesiana sede Guayaquil, en las diversas asignaturas relacionadas con el control automático, Instrumentación y domótica que se imparten a lo largo de la malla curricular.

1.9 Descripción de la propuesta

Se implementa un conjunto de kits para el diseño de aplicaciones de sistemas embebidos basados en tecnología RASPBERRY-PI y ODROID-U3 adecuados para la elaboración y desarrollo de prácticas en diversas asignaturas de la carrera de ingeniería electrónica.

La parte fundamental de estos kits son los computadores embebidos Raspberry-Pi y ODROID-U3 que trabajan en conjunto con dispositivos electrónicos diversos usados para controlar y operar equipos, máquinas, aparatos domésticos, equipos móviles, automóviles, instrumentos electrónicos y hasta plantas industriales.

Es de gran importancia para los estudiantes el manejo de sistemas y computadores embebidos que permitan con facilidad establecer firmemente las habilidades de programación estrechamente relacionadas con los conceptos y nociones del control industrial.

Las placas Raspberry- Pi y ODROID-U3 son maravillas en miniatura, aguardan en su interior un importante poder de cómputo en un tamaño no más grande que el de una tarjeta de crédito. Son capaces de realizar cosas extraordinarias, que van desde el entretenimiento en el hogar hasta el control industrial.

No importa si el estudiante es principiante o si es un programador experto.

Raspberry-Pi y ODROID-U3 son herramientas excelentes para desarrollar habilidades como programador. Ideales para proyectos de control, domótica, automatización y robótica.

Es por este motivo que se utilizan estas placas para implementar los Kits didácticos de laboratorio para el desarrollo de aplicaciones con sistemas embebidos.

1.10 Beneficiarios

Los beneficiarios directos de este trabajo son los estudiantes de Ingeniería Electrónica de la Universidad Politécnica Salesiana, sede Guayaquil, los cuales

cuentan con kits de elementos acorde con el avance de la tecnología para realizar prácticas de laboratorio.

1.11 Impacto

Estos kits y el desarrollo de sus aplicaciones, mejoran el desempeño de los estudiantes permitiéndoles obtener un panorama más amplio a los conceptos impartidos en las aulas de clases y laboratorios.

Con los mismos se pueden implementar de manera práctica diversos conceptos y fundamentos teóricos que ayudan a la comprensión y fortalecimiento del aprendizaje.

CAPÍTULO 2

MARCO TEÓRICO

2.1 HARDWARE

2.1.1 ¿Qué es Raspberry-pi?

Raspberry Pi es un ordenador de placa reducida o (placa única) de bajo coste, desarrollado en Reino Unido por la Fundación Raspberry Pi y Ramón Santamaría "RAYSAN", con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

Es una maravilla en miniatura, aguarda en su interior un importante poder de cómputo en un tamaño no más grande que el de una tarjeta de crédito. Es capaz de realizar cosas extraordinarias, puede ser utilizada en proyectos de electrónica, y para muchas de las cosas que hace el PC de escritorio, como hojas de cálculo, procesadores de texto y juegos. También reproduce vídeo de alta definición. Se puede ver que está siendo utilizado por los estudiantes de todo el mundo para aprender a programar.

El diseño incluye un “System-on-a-chip Broadcom BCM2835”, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz (el firmware incluye unos modos “Turbo” para que el usuario pueda hacerle overclock de hasta 1 GHz sin perder la garantía), un procesador gráfico (GPU) VideoCore IV, y 512 MB de memoria RAM (aunque originalmente al ser lanzado eran 256 MB). El diseño no incluye un disco duro ni unidad de estado sólido, ya que usa una tarjeta SD para el almacenamiento permanente; tampoco incluye fuente de alimentación ni carcasa.

2.1.2 Tipos de Raspberry Pi

En la actualidad existen varios modelos de computadores embebidos Raspberry-Pi entre los que se puede citar:

- Raspberry-Pi modelo A
- Raspberry-Pi modelo A+
- Raspberry-Pi modelo B
- Raspberry-Pi modelo B+
- Raspberry-Pi 2 modelo B

2.1.2.1 Raspberry Pi Modelo A

Las ventas iniciales fueron del modelo A. El modelo A solo tiene un puerto USB, carece de controlador Ethernet y cuesta menos que el modelo B, el cual tiene dos puertos USB y controlador Ethernet 10/100.

A pesar que el Modelo A no tiene un puerto RJ45, se puede conectar a una red usando un adaptador USB-Ethernet suministrado por el usuario.

Por otro lado, a todos los modelos se puede conectar un adaptador Wi-Fi por USB, para tener acceso a redes inalámbricas o internet. El sistema cuenta con 256 MB de memoria RAM en su modelo A, y con 512 MB de memoria RAM en su modelo B.

Como es típico en los ordenadores modernos, se pueden usar teclados y ratones con conexión USB compatible con Raspberry Pi.

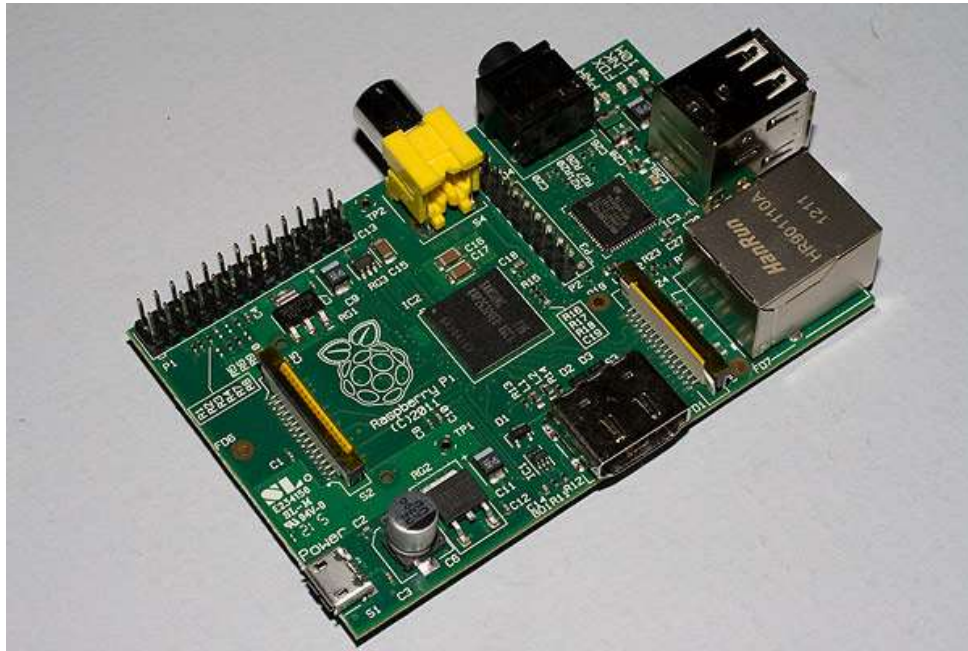


Figura 2.1 Raspberry Pi Modelo A

Fuente: Retro Kit (2011). Recuperado de <http://www.retro-kit-co-uk/user/custom/Raspi/images/RaspPi-sml.jpg>

2.1.2.2 Raspberry Pi Modelo A+

El modelo A+ está derivado del A, con el cual comparte gran parte de las especificaciones, pero cuenta con algunas mejoras. Acepta tarjetas Micro SD y el diseño es más estético, rebajando el tono industrial del dispositivo precedente.

La Raspberry Pi A+ es el modelo más barato que ofrece hoy la fundación. A este precio se ofrece un SoC Broadcom BCM2835 con 256 MB de RAM. No es un dato menor, pues uno de los objetivos de la organización es hacer llegar hardware libre asequible a segmentos de la población que no se puedan permitir pagar los precios del mercado. Por eso el dispositivo ha sido útil para sectores como la educación.

Otra de las características de Raspberry Pi A+ es su capacidad de ahorro energético. El nuevo modelo consume un 30% menos de electricidad que la versión B+. Eso convierte al dispositivo en una herramienta ideal para llevar a cabo proyectos donde el ahorro de energía es una pieza central.



Figura 2.2 Raspberry Pi Modelo A+

Fuente: Blogthinkbig (2014). Recuperado de <http://blogthinkbig.com/wp-content/uploads/2014/11/raspberry-pi-a1-620x412.jpg>

Las mejoras implementadas en la nueva versión A+ pretenden llamar la atención del público potencial de Raspberry Pi, tratando de dar un impulso a la línea A, que ha vendido 100.000 unidades hasta ahora. En cambio, entre los modelos B y B+ han colocado en el mercado casi cuatro millones de dispositivos.

En realidad parte del éxito masivo de la fundación se debe a esta última línea, algo que puede cambiar con Raspberry Pi A+, sobre todo debido al ahorro energético y al precio. (<http://blogthinkbig.com/asi-es-la-nueva-raspberry-pi/>)

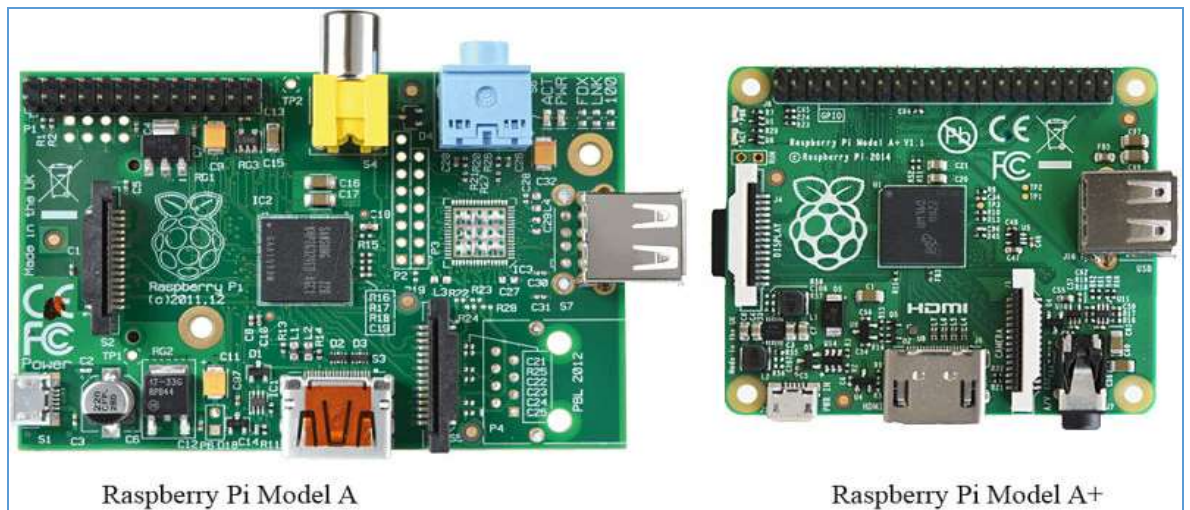


Figura 2.3 Diferencias entre modelos

Fuente: Into Robotics (2014). Recuperado de http://www.intorobotics.com/wp-content/uploads/2014/11/raspberry-pi-model-a-vs-mod_opt.jpg

2.1.2.3 Raspberry Pi Modelo B

En realidad, se trata de una diminuta placa base de 85 x 54 milímetros (del tamaño aproximado de una tarjeta de crédito) en el que se aloja un chip Broadcom BCM2835 con procesador ARM hasta a 1 GHz de velocidad (modo Turbo haciendo overclock), GPU VideoCore IV y 512 Mbytes de memoria RAM (Las primeras placas contaban con sólo 256MB de RAM).

En cuanto a la conexión de red, se dispone de un puerto Ethernet (los modelos A y A+ no disponen de puerto Ethernet) para enchufar un cable RJ-45 directamente al router o se puede recurrir a utilizar cualquier adaptador inalámbrico WiFi compatible. En este caso, eso sí, conviene que elogiar por la Raspberry Pi que incorpora dos puertos USB, ya que de lo contrario, no se podrá conectar el teclado y el ratón.

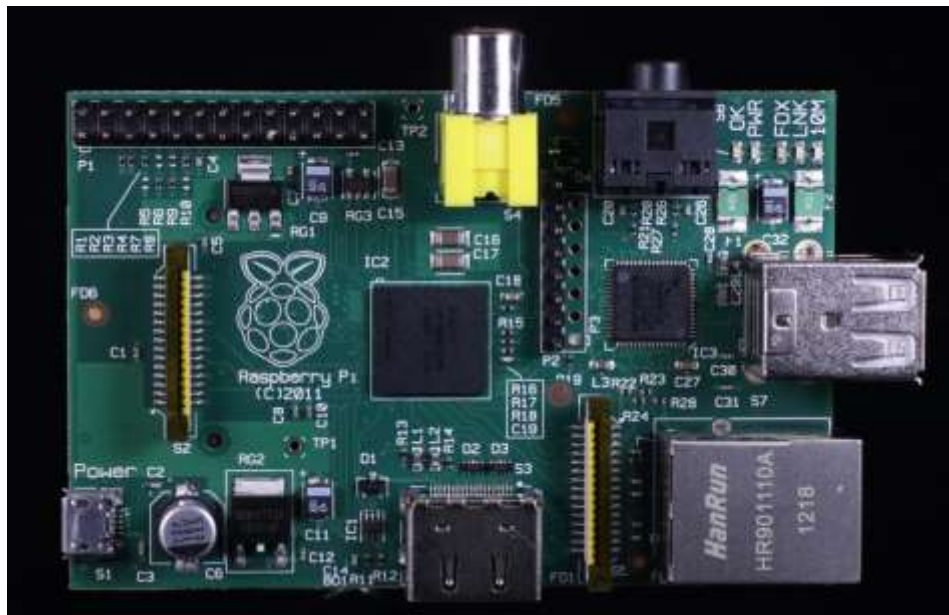


Figura 2.4 Modelo B

Fuente: Elinux.org (2012). Recuperado de <http://elinux.org/File:RPi-Front-JPB.jpg>

El Raspberry Pi no viene con reloj en tiempo real, por lo que el sistema operativo debe usar un servidor de hora en red, o pedir al usuario la hora en el momento de arrancar el ordenador. Sin embargo se podría añadir un reloj en tiempo real (como el DS1307) con una batería mediante el uso de la interfaz I²C.

El 5 de septiembre de 2012, se anunció una revisión 2.0 de la placa, que ofrecía un pequeño número de correcciones y mejoras, como unos agujeros de montaje, un circuito para hacer reset, soporte para depuración JTAG, etc.

El 15 de octubre de 2012, la fundación anunció que todos los Raspberry Pi Modelo B serían enviados a partir de ese momento con 512 MiB de RAM en vez de 256 MB. Los esquemas del modelo A y el modelo B fueron lanzados el 20 de abril de 2012 por la fundación. (http://es.wikipedia.org/wiki/Raspberry_Pi)

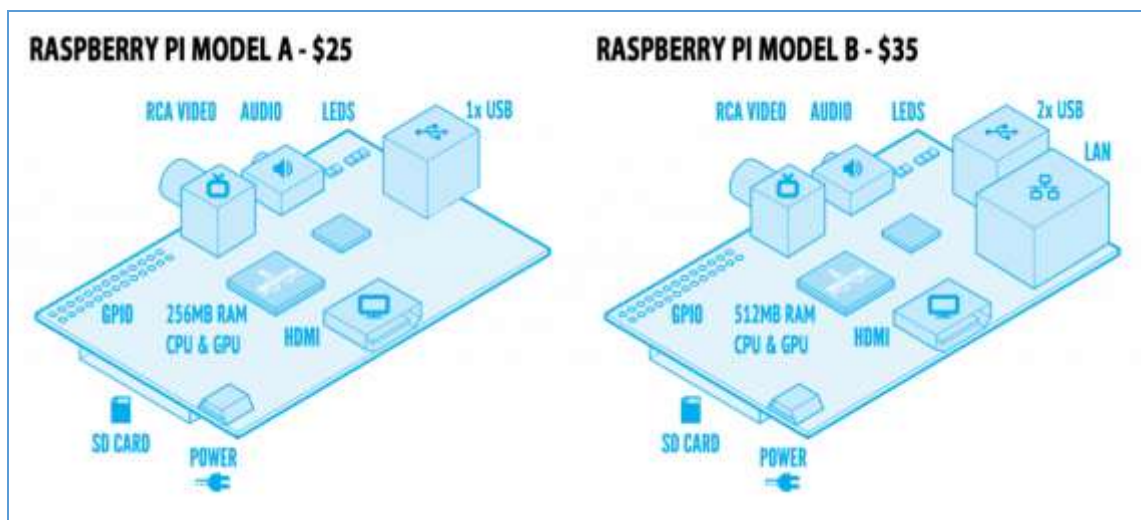


Figura 2.5 Características & costos

Fuente: Hack Things (2013). Recuperado de

http://d19vp51bag8frf.cloudfront.net/wp-content/uploads/2013/03/raspberry_pi_comparison-1500x663.png

2.1.2.4 Raspberry Pi Modelo B +

Raspberry Pi B+ no es un gran salto respecto a su predecesor, únicamente mejora algunas especificaciones manteniendo el formato y la mayoría de componentes del modelo B.

Estas son sus especificaciones:

- Procesador: Broadcom BCM2835 SoC full HD (igual que el modelo B)
- RAM: 512 MB SDRAM 400 MHz (igual que el modelo B)
- Almacenamiento: tarjeta microSD (en el modelo B era una SD)
- USB: cuatro puertos USB 2.0 (en el modelo B sólo había 2)

- Energía: 600mA hasta 1.8A a 5V (en el modelo B, 750mA hasta 1.2A a 5V)
- Pines GPIO: 40 (en el modelo B, 26)



Figura 2.6 Modelo B+

Fuente: I.stack.imgur (2013). Recuperado de <http://i.stack.imgur.com/Hv465.jpg>

¿Cuál es la diferencia entre el modelo B y B +?

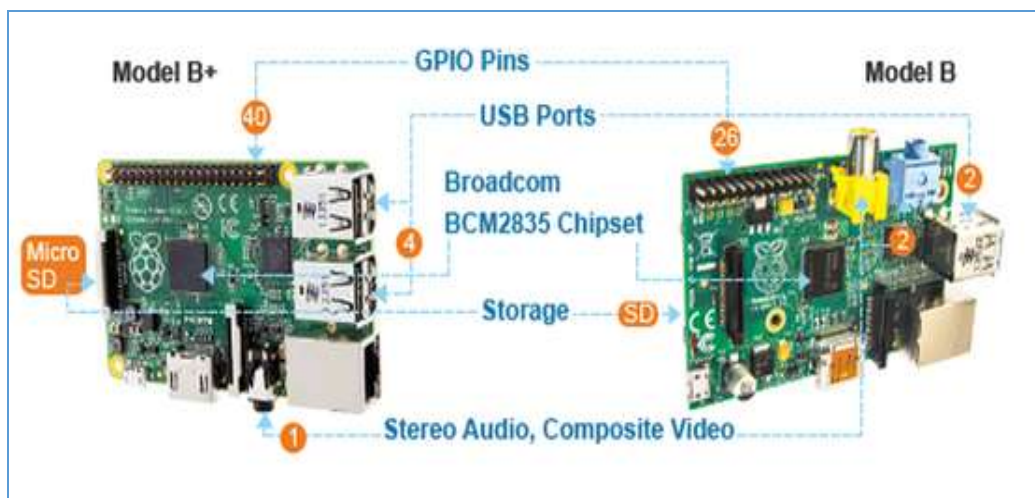


Figura 2.7 Diferencias Modelo B & Modelo B+

Fuente: I.stack.imgur (2013). Recuperado de <http://i.stack.imgur.com/9Lac3.png>

Cuatro (4) puertos USB - por primera vez, se puede tener un teclado, el ratón y el dongle wifi conectado sin necesidad de un concentrador con alimentación. Se debe tener en cuenta, que se necesita una fuente de alimentación más potente para el B + si se desean utilizar los 4 puertos.

Mejor capacidad de conexión en caliente USB, se pueden conectar los dispositivos de seguridad wifi en caliente sin reinicios.

Un nuevo puerto Ethernet con luces de actividad.

Los puertos USB y Ethernet están ahora en línea recta, en lugar de en la disposición escalonada de la B.

El puerto de vídeo compuesto amarillo se ha suprimido ahora hay un conector de 3,5 mm combinado que proporciona la señal de audio y el vídeo analógico.

El puerto HDMI, el DSI (display) puertos CSI (cámara) y todos han sido movido ligeramente.

La toma de micro corriente USB ha sido cambiada de lugar. Esto significa que todos los puertos están en el mismo lado de la Pi, por lo que la gestión de conexionado de cables es mucho mejor con el B +.

El B + dispone de una cabecera GPIO de 40 pines (en comparación con el de la B que es de 26) y ya no tiene una cabecera P5 y P3.

Cuatro (4) agujeros de montaje, convenientemente ubicados en las esquinas de la placa en lugar de los 2 orificios colocados extrañamente en la B.

En la parte posterior, el B + utiliza una tarjeta micro SD (en comparación con el tamaño completo de la tarjeta SD de la B) con una acción push-push (es decir, se empuja y se encierra en su lugar, usted lo empuja de nuevo y sale). Menores requisitos de energía - se utilizarán entre 0.5 watts y 1 watt menos que el modelo B.

Mucho mejor salida de audio gracias a una fuente de alimentación dedicada

2.1.2.5 Raspberry-Pi 2 Modelo B

La Raspberry Pi 2 Modelo B tiene idéntico tamaño que el modelo B+, presentado en el apartado anterior, pero se caracteriza por tener una mayor potencia.

El nuevo modelo ofrece un procesador de cuatro núcleos a 900MHz y 1GB de RAM, mientras que el B+ posee uno con un rendimiento seis veces menor y 512MB de RAM.

El resto de las características son idénticas: 4 puertos USB, HDMI, Ethernet, ranura para microSD y conector de audio.

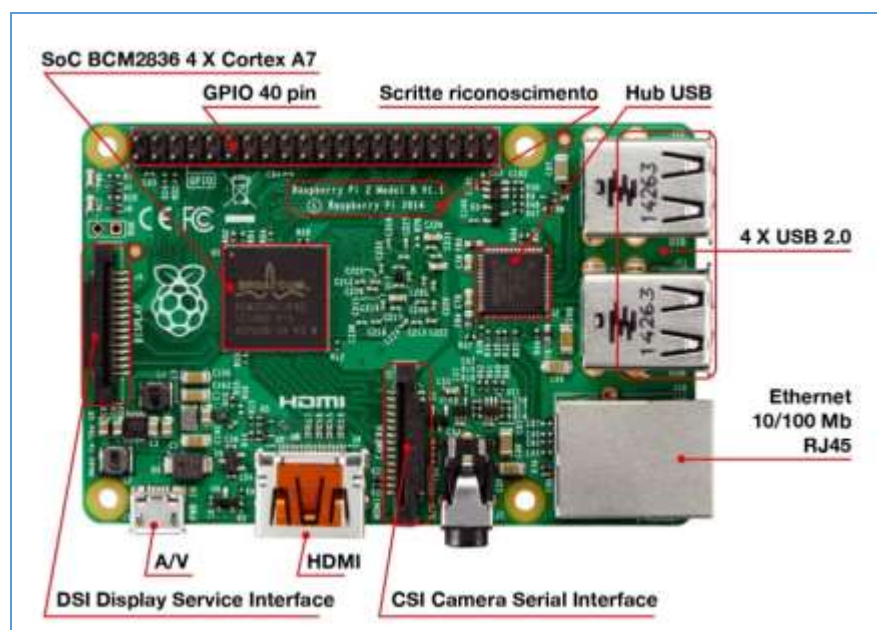


Figura 2.8 Raspberry Pi2 Modelo B

Fuente: Tienda Raspberry Pi (2012). Recuperado de <http://www.raspberrypi.com/images/raspberry-pi-2-placa.jpg>

Pensada para el sector educativo, la Raspberry Pi se convirtió prácticamente en un objeto de culto entre los fanáticos de la informática.

Los alcances de este equipo podrían sin embargo incrementarse gracias a una alianza con Microsoft, que ofrecerá una versión especial de Windows 10 para la Raspberry Pi 2 Modelo B.

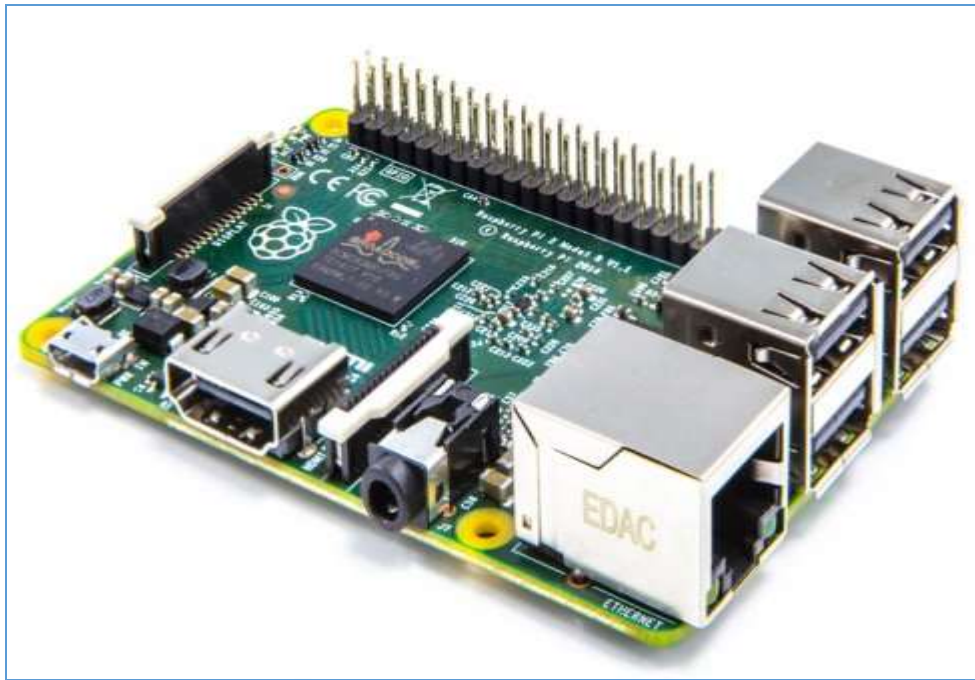


Figura 2.9 En perspectiva Raspberry Pi2 Modelo B

Fuente: Infobae (2013). Recuperado de

<http://cdn01.ib.infobae.com/adjuntos/162/imagenes/012/160/0012160439.jpg?0000-00-00-00-00>

El ordenador Raspberry Pi modelo B original salió a la venta en febrero de 2012 y estableció un nuevo estándar, destruyendo el dominio del PC en el mercado del hogar y la educación. Desde entonces, se han distribuido por todo el mundo más de 3 millones de placas en sus diferentes formatos: A, B, A+, B+ y Modelo 2.

El concepto original de Raspberry Pi era el de una placa de ordenador de alto rendimiento que ofreciera acceso a Internet con gráficos HD a precio muy bajo. Las placas ofrecen una plataforma para niños y adultos con cualquier nivel de conocimientos previos para adquirir conocimientos sobre la ciencia informática y ayudar a desarrollar la red informática mundial (World-Wide-Web) y el Internet de las Cosas del futuro.

A pesar de que las placas Raspberry Pi se diseñaron principalmente para fines educativos, han ido adquiriendo más popularidad entre los fabricantes de sistemas integrados. Para poder satisfacer a este mercado, Raspberry Pi Foundation se ha encargado de asegurar la compatibilidad con versiones anteriores en cada nueva

versión. El módulo Compute de estructura desnuda está destinado específicamente al fabricante OEM.

2.1.3 De qué está hecha la Raspberry Pi tipo B

Con unas dimensiones de placa de 8.5 por 5.3 cm, en el modelo B de la Raspberry Pi, que es el que se utiliza en las prácticas contenidas en este trabajo de titulación, se encuentran unas características muy interesantes.

En su corazón se encuentra con un chip integrado Broadcom BCM2835, que contiene un procesador ARM11 con varias frecuencias de un procesador gráfico VideoCore IV, y distintas cantidades de memoria RAM que se pueden asignar al proceso gráfico.

Se encuentra además una salida de vídeo y audio a través de un conector HDMI, con lo que se consigue conectar la tarjeta tanto a televisores como a monitores que cuenten con dicha conexión. En cuanto a vídeo se refiere, también cuenta con una salida de vídeo compuesto y una salida de audio a través de un minijack.

Un SoC (system on chip) es un integrado que incorpora todos los componentes del sistema. En el caso de la Raspberry B, lleva un Broadcom BCM2835 que incluye: el procesador (ARM1176JZF-S), la tarjeta gráfica con aceleración gráfica 3D y de video en alta definición, 512 Mb de RAM, tarjeta de sonido estéreo y bus USB.

2.1.3.1 Tarjeta de red

Se tiene a disposición un conector RJ-45 conectado a un integrado lan9512 -jzx de SMSC que proporciona conectividad a 10/100 Mbps además de buses USB 2.0.

2.1.3.2 Conexiones

Se dispone de:

- Dos buses USB.
- Puerto ethernet RJ-45.
- Salida analógica de audio estéreo por jack de 3.5 mm.
- Salida digital de video + audio HDMI.
- Salida analógica de video RCA.
- Pines de entrada y salida de propósito general. (GPIO).

2.1.4 Especificaciones Raspberry Pi tipo B

Tabla 2.1: Especificaciones Raspberry Pi

	Modelo A	Modelo B	Modelo B +
Precio objetivo:	US \$ 25 impuestos Ext (GBP £ 16 SIN IVA)	US \$ 35 impuestos Ext (GBP £ 22 SIN IVA)	US \$ 35 impuestos Ext (GBP £ 22 SIN IVA)
System-on-a-chip (SoC):	Broadcom BCM2835 (CPU + GPU. SDRAM es un chip separado apilados en la parte superior)		
UPC:	700 MHz ARM11 ARM1176JZF-S núcleo		
GPU:	Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1080p30 H.264 de alto perfil de codificación / decodificación		
Memoria (SDRAM) iB	256 MiB (previsto con 128 MiB, actualizado a 256 MiB el 29 feb 2012)	256 MiB (hasta 15 de octubre 2012); 512 MiB (desde 15 Octubre 2012)	512 MiB
Puertos USB 2.0:	1 (proporcionado por el BCM2835)	2 (a través de un concentrador USB integrado)	4 (a través de un concentrador USB intergrated)
Salidas de vídeo:	Vídeo compuesto RCA compuesto, HDMI (no al mismo tiempo)		Vídeo compuesto requiere adaptador Polo 4
Salidas de audio:	Conector TRS jack 3,5 mm, HDMI		
Entradas de audio:	ninguna, pero un micrófono USB o tarjeta de sonido podrían añadirse		

Almacenamiento a bordo:	Secure Digital ranura para tarjetas SD / MMC / SDIO		Micro Secure ranura Digital / MicroSD
A bordo de red:	Ninguno	10/100 con cables Ethernet RJ45	
Periféricos de bajo nivel:	26 / salida (GPIO) pines de entrada Propósito General, Serial Peripheral Interface Bus (SPI), I ² C, I ² S ^[2] , Universal asíncrono receptor / transmisor (UART)		40 / salida (GPIO) pines de entrada Propósito General, Serial Peripheral Interface Bus (SPI), I ² C, I ² S, ^[2] I ² C IDC Pins, universal asíncrono receptor / transmisor (UART)
Reloj en tiempo real:	Ninguno		
Los valores de potencia:	300 mA, (1,5 W) ^[1]	700 mA, (3,5 W)	~ 650 mA, (3,0 W) ^[3]
Fuente de alimentación:	5 V (DC) a través de Micro USB de tipo B o encabezado GPIO		
Tamaño:	85,0 x 56,0 mm x 15 mm	85,0 x 56,0 mm x 17 mm	85,0 x 56,0 mm x 17 mm
Peso:	31g	40g	40g

Nota: Elinux.org (2012). Fuente: http://elinux.org/RPi_Hardware

2.1.5 ARQUITECTURA

2.1.5.1 ARM vs. x86

El procesador en el interior de la Raspberry Pi modelo B es un procesador multimedia Broadcom BCM2835 system-on-chip (SoC). Esto quiere decir que la mayor parte de los componentes del sistema, incluidos la unidad central de procesamiento y la de gráficos junto con el audio y el hardware de comunicaciones, se encuentran integrados dentro de aquel único componente oculto ubicado justo debajo del chip de la memoria de 256 MB en el centro de la placa. No es sólo el

diseño del SoC lo que hace al BCM2835 diferente del procesador de un PC o laptop. Lo que lo hace también diferente es que utiliza una arquitectura de conjunto de instrucciones (Instruction Set Architecture, ISA) distinta, conocida como ARM.

Desarrollada por Acorn Computers años atrás a finales de 1980, la arquitectura ARM es relativamente poco conocida en el mundo de las computadoras de escritorio. En donde destaca, sin embargo, es en los dispositivos móviles: en los teléfonos de bolsillo es casi seguro que se encuentre al menos un núcleo de procesamiento basado en ARM escondido en su interior. La combinación de la arquitectura RISC (Simple Reduced Instruction Set) y su bajo consumo energético lo convierten en la opción perfecta frente a los chips de computadoras de escritorio que demandan altos consumos y arquitecturas CISC (Complex Instruction Set).

El BCM2835 basado en ARM es el secreto que explica cómo la Raspberry Pi es capaz de funcionar con tan sólo una fuente de alimentación de 5V 1A suministrada por el puerto micro-USB a bordo. Es también la razón por la cual no encontrará ningún disipador térmico sobre el dispositivo: el bajo consumo de energía del chip se traduce directamente en muy poco calor residual, incluso durante las tareas de procesamiento más complejas.

Esto, sin embargo, significa que la Raspberry Pi no es compatible con el software de las PC tradicionales. La mayoría del software para computadoras de escritorio y laptops se construye teniendo en cuenta la arquitectura de conjunto de instrucciones x86, presente en los procesadores como AMD, Intel y VIA. Por consiguiente, este software no funciona en la Raspberry Pi que se basa en la arquitectura ARM.

El BCM2835 utiliza una generación del diseño del procesador ARM conocida como ARM11, que a su vez está diseñada en torno a una versión de la arquitectura de conjunto de instrucciones conocida como ARMv6. Vale la pena recordar que ARMv6 es una arquitectura ligera y potente, pero tiene un rival en la arquitectura más avanzada, ARMv7 utilizada por la familia de procesadores ARM Cortex. El software desarrollado para la ARMv7, al igual que el desarrollado para la x86, es por desgracia incompatible con el BCM2835 de la Raspberry Pi, aunque los desarrolladores generalmente pueden convertir el software para adecuarlo.

Esto no quiere decir que va a estar limitado en sus opciones. Como se verá más adelante, hay un montón de software disponible para el conjunto de instrucciones ARMv6, y en tanto que la popularidad de la Raspberry Pi siga aumentando, el software disponible seguirá creciendo.

2.1.5.2 Windows vs. Linux

Otra diferencia importante entre la Raspberry Pi y el PC de escritorio o laptop, aparte de su tamaño y su costo, es el sistema operativo (el software que permite controlar al computador) que utiliza.

La mayoría de las PCs y laptops disponibles hoy en día funcionan con alguno de estos dos sistemas operativos:

Microsoft Windows o Apple OS X. Ambas plataformas son de código cerrado, creados en un ambiente reservado utilizando técnicas patentadas.

Estos sistemas operativos son conocidos como de código cerrado por la naturaleza de su código fuente, es decir, la receta en lenguaje de computadora que le dice al sistema que hacer. En el software de código cerrado, esta receta es mantenida como un secreto muy bien guardado. Los usuarios pueden obtener el software terminado, pero nunca ver cómo está hecho.

La Raspberry Pi, por el contrario, está diseñada para ejecutar un sistema operativo llamado GNU/Linux (de ahora en adelante se refiere a éste simplemente como Linux). A diferencia de Windows u OS X, Linux es de código abierto. Esto quiere decir que es posible descargar el código fuente del sistema operativo por completo y hacer los cambios que uno desee. Nada es ocultado, y todos los cambios hechos están a la vista del público. Este espíritu de desarrollo de código abierto ha permitido a Linux rápidamente ser modificado para poder ejecutarse sobre la Raspberry Pi, un proceso conocido como portabilidad. Al momento de escribir estas líneas, varias versiones de Linux (conocidas como distribuciones) han sido portadas al chip BCM2835 de la Raspberry Pi, incluyendo Debian, Fedora Remix y Arch Linux.

Las distintas distribuciones atienden diferentes necesidades, pero todas ellas tienen algo en común: son de código abierto. Además, por lo general, todas son compatibles

entre sí: el software escrito en un sistema Debian funcionará perfectamente bien en uno con Arch Linux y viceversa.

Linux no es exclusivo de la Raspberry Pi. Cientos de diversas distribuciones están disponibles para PCs de escritorio, laptops e incluso dispositivos móviles; la popular plataforma Android de Google está desarrollada en encima de un núcleo Linux. Si disfruta la experiencia de utilizar Linux en la Raspberry Pi, podría considerar añadir otros dispositivos informáticos suyos para que lo utilicen también. Estos coexistirán felizmente con su sistema operativo actual, permitiéndole disfrutar de las ventajas de ambos sistemas mientras le brindan un entorno familiar cuando su RasPi no se encuentre disponible.











Igual que con la diferencia entre la arquitectura ARM y la x86, hay un punto clave que hace la diferencia práctica entre Windows, OS X y Linux: el software escrito para Windows u OS X no funciona en Linux. Afortunadamente, hay un montón de alternativas compatibles para la gran mayoría de los productos de software comunes y lo mejor, casi todos son de libre uso y de código abierto como lo es el propio sistema operativo.












2.1.6 Software para Raspberry Pi











Desde aquí se podrá descargar casi todo el software disponible para la Raspberry Pi, desde sistemas operativos, instaladores, juegos, aplicaciones, etc. Las instrucciones para su instalación están descritas en cada página web.

Tabla 2.2: Lista de software que pueden ser instalados en su embebido Raspberry

Descargas de Software para Raspberry Pi				
Proyecto	Fecha	Página WEB	Categoría	Sistema Operativo

Raspbian	25- SEP- 2013	<u>Raspberry Pi Downloads</u>	Sistema Operativo	
Soft-float Debian "wheezy"	08- AGO- 2012	<u>Raspberry Pi Downloads</u>	Sistema Operativo	
Arch Linux ARM	13- AGO- 2013	<u>Raspberry Pi Downloads</u>	Sistema Operativo	
RISC OS	10- JUL- 2013	<u>Raspberry Pi Downloads</u>	Sistema Operativo	
FreeBSD	01- FEB- 2013	<u>FreeBSD</u>	Sistema Operativo	
openSUSE	10- ENE- 2013	<u>opensuse.org</u>	Sistema Operativo	
Occidentalis	31- AGO- 2012	<u>Adafruit learning System</u>	Sistema Operativo	
pisces image	10- JUL- 2012	<u>Mike Thompson</u>	Sistema Operativo	
Plan 9	30- NOV- 2012	<u>Bendyworks</u>	Sistema Operativo	
Gentoo Linux	29- SEP- 2012	<u>intelminer.com</u>	Sistema Operativo	

Hexxeh		<u>Hexxeh Images</u>	Sistema Operativo	
Puppi	16-FEB-2013	<u>Puppi School</u>	Sistema Operativo	
PiLFS	16-DIC-2012	<u>Linux From Scratch</u>	Sistema Operativo	
Pidora Fedora 18 Remix	09-AGO-2013	<u>Fedora Remix</u>	Sistema Operativo	
RaspBMC	02-OCT-2013	<u>Raspberry Pi Downloads</u>	Sistema Operativo	
Win32 Disk Imager	04-JUN-2013	<u>sourceforge</u>	Instalador SO	
RMPrepUSB	26-JUN-2013	<u>RMPrepUSB.com</u>	Instalador SO	
BerryBoot	13-FEB-2013	<u>BerryTerminal</u>	Instalador SO	
NOOBS V1.3	28-SEP-2013	<u>Raspberry Pi downloads</u>	Instalador SO	
pwnpi	14-DIC-2012	<u>pwnpi</u>	Seguridad	
ha-pi	06-JUL-	<u>sourceforge</u>	Seguridad	

	2012			
Raspbmc	AGO 2013	<u>Raspbmc</u>	XBMC Media Center	
OpenELEC	14- SEP- 2013	<u>OpenELEC</u>	XBMC Media Center	
RasPlex	24- MAR- 2013	<u>RasPlex</u>	XBMC Media Center	
SqueezePlug	16- FEB- 2013	<u>SqueezePlug.eu</u>	XBMC Media Center	
Pi MusicBox	24- JUN- 2013	<u>Pi MusicBox</u>	XBMC Media Center	
RPi-TC	06- FEB- 2013	<u>Raspberry Pi Thin Client project</u>	Dispositivo	
OpenEnergyMonitor	23- NOV- 2012	<u>OpenEnergyMonitor</u>	Dispositivo	
uelastix	26- JUN- 2013	<u>uelastix</u>	VOIP	
Incredible Pi	27- NOV- 2012	<u>Incredible Pi</u>	VOIP	
raspbx	19- ENE- 2013	<u>raspbx</u>	VOIP	

FishPi	16- FEB- 2013	<u>FishPi</u>	Proyecto	
Android Pi	16- FEB- 2013	<u>Android Pi wiki</u>	Escritorio	
RetroPie	10- FEB- 2013	<u>petRockBlog</u>	Juego	
Quake III		<u>Electrolitoblog</u>	Juego	
PiMAME	30- OCT- 2013	<u>Shea Silverman's Blog</u>	Juego	

Nota: Tienda Raspberry Pi (2012). Fuente: <http://www.raspberrypi.org/software-raspberry-pi.php>

La tabla anterior muestra el software que podrían ser instalados en su embebido Raspberry, esta lista dará una mejor guía de las aplicaciones y bondades de la Raspberry.

2.1.7 ACCESORIOS

La Raspberry PI tiene muchos tipos de accesorios y extensiones dependiendo del propósito en el cual se la utiliza. La mayoría de los accesorios genéricos existentes para PCs en el mercado son compatibles con raspberry-Pi, sin embargo en la siguiente dirección web: http://elinux.org/RPi_VerifiedPeripherals, se puede encontrar un detalle completo de los accesorios oficialmente soportados por la Raspberry-pi.

Como Mini Ordenador lo más básico que de primera mano se utiliza para su puesta en marcha y funcionamiento es lo siguiente:

- 1 x Adaptador de corriente 5V/ 1 A USB
- 1 x Cable de red Ethernet CAT5 (1m)
- 1 x Cable de vídeo HDMI (1,8m)
- 1 x Memoria MicroSD 8GB Clase 10 Precargada con el Sistema Operativo a utilizar.
- 1 x Teclado USB.
- 1 x Mouse USB.
- 1 x MODEM WIFI (opcional)
- 1 x Hub USB (opcional).
- 1 x Monitor HDMI o Video Compuesto (Tv).



Figura 2.10 Accesorios que complementan la Raspberry

Fuente: Elinux.org (2012). Recuperado de http://elinux.org/RPi_Hardware

2.1.8 ¿Qué es ODROID?

Odroid (Open Android) es una plataforma abierta de hardware pensada para desarrolladores Android y Linux que ofrece una variedad de placas de desarrollo de tamaño reducido y gran potencia, entre las que se destacan el Odroid-X3, Odroid-U3, Odroid-XU y Odroid-C1. Estas placas poseen el desempeño cercano a una PC regular y pueden correr el sistema operativo Android y varias otras distribuciones de Linux, como por ejemplo: Ubuntu, Debian, ArchLinux, OpenSUSE, Fedora, etc.

En las prácticas realizadas en este trabajo de titulación se utiliza la placa Odroid-U3 basada en el procesador Samsung de 1.7GHz Exynos4412 Primer Cortex-A9 de cuatro núcleos, con 2GB de memoria RAM. Esta placa Odroid en conjunto con el módulo de entradas y salidas IO shield proporciona múltiples entradas y salidas de propósito general (GPIO) además de varias entradas analógicas, por lo cual es de interés especial por las mayores posibilidades de interfaz que ofrece para la conexión de dispositivos externos para aplicaciones de Linux embebido en áreas como: robótica, monitoreo, control industrial y doméstico, etc.

Para las prácticas se usa el sistema operativo Ubuntu 14.04.1 en la placa debido a que, junto con Debian y ArchLinux, son posiblemente los más livianos, versátiles y por tanto ideales para sistemas embebidos, sin embargo para nuevos usuarios Ubuntu es mucho más fácil de manejar que ArchLinux. Aunque estas prácticas se basan en el Odroid-U3, casi toda la información presentada debería aplicarse también al uso de las otras placas Odroid.

2.1.8.1 Tipos de Odroid

En la actualidad existen varios modelos de computadores embebidos ODROID entre los más versátiles y utilizados para nuestros propósitos se puede citar:

- ODROID-U3
- ODROID-XU3
- ODROID-C1.

2.1.8.1.1 ODROID-U3

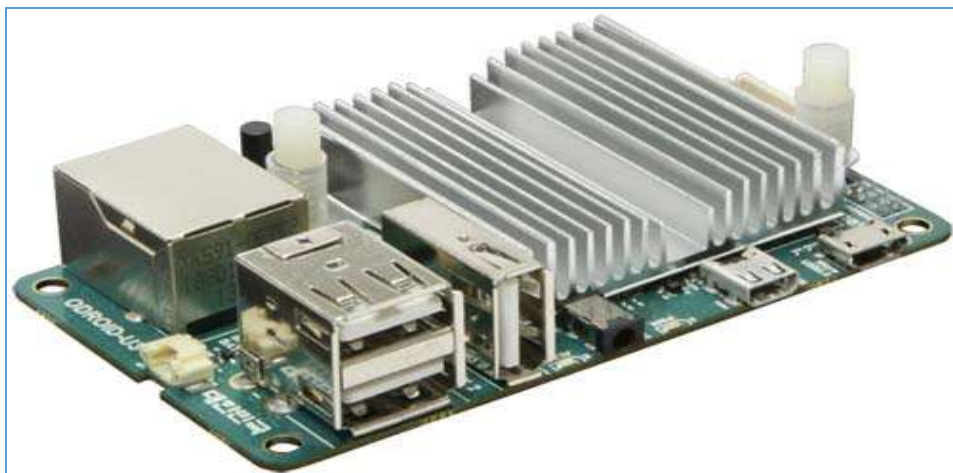


Figura 2.11 Odroid U3

Fuente: dn.odroid (2014). Recuperado de
<http://dn.odroid.com/homebackup/ODROID-U3.jpg>

Ordenador Linux Potente con las principales características:

- Procesador de 1.7GHz Quad-Core y 2GByte de memoria RAM.
- Connector Ethernet 10 / 100Mbps Jack RJ-45 LAN.
- Host de 3 puertos USB 2.0 de alta velocidad.
- Códec de audio con conector de auriculares incorporado.
- Sistema Operativo Xubuntu 13.10 o sistema operativo Android 4.x
- Tamaño: 83 x 48 mm, Peso: 48g incluido disipador de calor.

2.1.8.1.2 ODROID-XU3

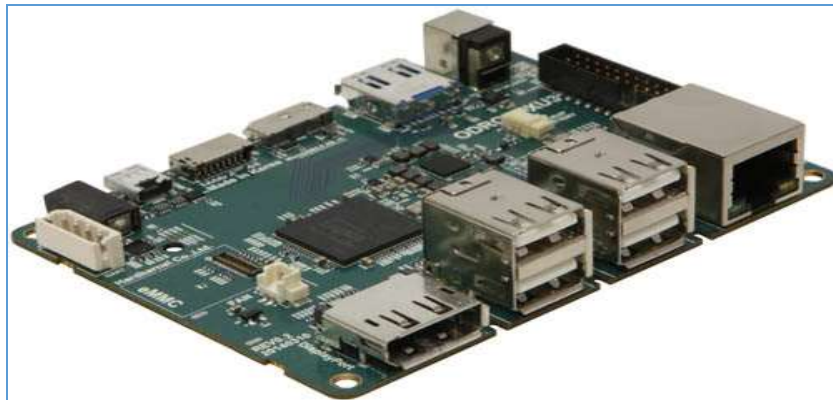


Figura 2.12 Odroid XU3

Fuente: dn.odroid (2014). Recuperado de
<http://dn.odroid.com/homebackup/ODROID-XU3.jpg>

El ordenador Linux Multi-Proceso Heterogéneo (HMP) con las siguientes características principales:

- Procesador Samsung Cortex-A15 Exynos5422 Quad-core 2.0Ghz y Cortex-A7 de cuatro núcleos
- Conector Macho-T628 MP6 (OpenGL ES 3.0 / 2.0 / 1.1 y OpenCL 1.1 Perfil completo).
- Memoria de arranque y almacenamiento Flash eMMC5.0 HS400.
- 4 Host USB: 1 USB 3.0 Host, 1 USB 3.0 OTG, 2 USB 2.0.
- Sistema Operativo Xubuntu 14.04 o Android 4.4.

- Tamaño: 90 x 70 x 18 mm.

2.1.8.1.3 ODROID-C1



Figura 2.13 Odroid C1

Fuente: dn.odroid (2014). Recuperado de <http://dn.odroid.com/homebackup/201412051744489785.jpg>

Considerada una potente computadora pequeña para propósitos generales, desarrollo de software o plataforma de desarrollo de proyectos ODROID-C1 le brinda un rendimiento increíble a un precio muy bajo.

Sus características principales son:

- Procesador de cuatro núcleos a 1.5Ghz Amlogic Cortex-A5 (ARMv7)
- Conector Macho-450 MP2 GPU (OpenGL ES 2.0 / 1.1 habilitado para Linux y Android)
- Memoria SDRAM incorporada de 1 Gbyte DDR3.
- Conexión Ethernet Gigabit
- 40 pines GPIOs
- Ranura eMMC4.5 HS200 flash Almacenamiento / ranura UHS-1 SDR50 tarjeta MicroSD.
- Host USB: 4 USB 2.0 Host, 1 USB OTG.

- Receptor de Infrarrojos (IR).
- Sistema Operativo Ubuntu 14.04 o Android KitKat.

2.1.9 ¿ De qué está hecha la ODROID U3 ?.

Con un tamaño de una tarjeta de crédito y su bajo coste, ODROID-U3 puede verse como un simple equipo para aficionados. Sin embargo, su rendimiento, software y facilidad de uso lo convierten en una excelente alternativa a un PC de elevado coste. Compatible con Android y Linux permite ejecutar los miles de programas disponibles de forma gratuita o que son de bajo costo. ODROID-U3 también ofrece muchas ventajas frente al típico ordenador con Windows u OSX: un funcionamiento silencioso, consumo medio de 5W y su portabilidad, puesto que cabe en el bolsillo de una camisa.

Su procesador es un Samsung Exynos 4412 Quad-core de 1.7GHz con 2 GB de RAM. Presenta unas dimensiones de sólo 83 x 48 mm y un peso es de 48 gr.

A diferencia de un PC, ODROID-U3 tiene la placa base, la memoria, la tarjeta de sonido e incluso el disipador de calor ensamblados. En lugar de usar un costoso disco duro como ocurren en un PC, el sistema operativo se ejecuta desde una pequeña y barata micro SD, como las utilizadas en las cámaras digitales. La tarjeta SD, así como el módulo eMMC (más rápido) puede ser solicitado junto con el ODROID-U3, que incluyen el popular sistema operativo Ubuntu ya instalado.

2.1.9.1 Características principales

- Procesador de 1.7GHz Quad-Core y 2GByte de memoria RAM.
- Conector Ethernet 10 / 100Mbps Jack RJ-45 LAN.
- Host de 3 puertos USB 2.0 de alta velocidad.
- Códec de audio con conector de auriculares incorporado.
- Sistema Operativo Xubuntu 13.10 o sistema operativo Android 4.x
- Tamaño: 83 x 48 mm, Peso: 48g incluido disipador de calor.

ODROID-U3 es adecuado para cualquier persona desde ingenieros de software profesionales que desarrollan proyectos de robótica hasta niños que aprenden a

programar con el lenguaje Scratch. Capaz de ejecutar los últimos sistemas operativos Ubuntu 13.10 y Android Jellybean, es idóneo para programar, aprender, jugar, como centro multimedia, servidor web, realizar trabajos de oficina y universidad, como plataforma de hardware E/S entre otras muchas aplicaciones. Su potente procesador de 1.7Ghz de cuatro núcleos, el bajo coste, el consumo eficiente de energía y la amplia biblioteca de software disponible hacen de ODROID-U3 un ordenador moderno y perfecto para trabajar o jugar.

2.1.9.2 Conociendo la Placa

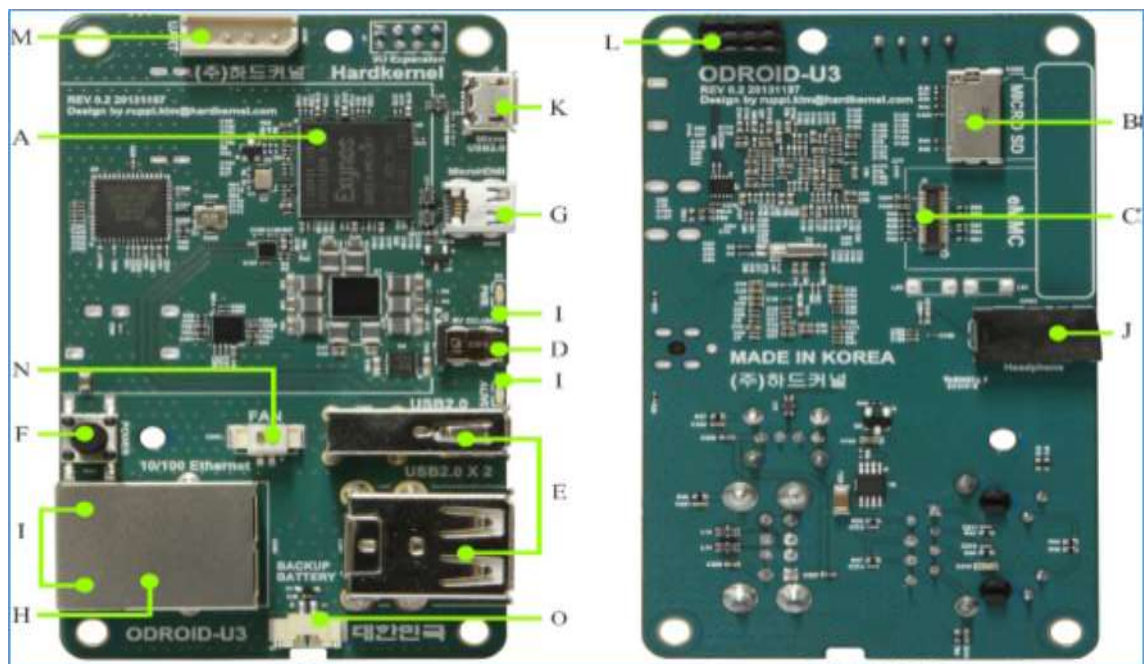


Figura 2.14 Partes placa Odroid U3

Fuente: Revista Odroid Magazine Volumen 1 Pág. 5 (2014)

El Procesador.- El corazón de ODROID-U3 presenta el mismo procesador que utiliza el teléfono Samsung Galaxy S3. Se trata de un sistema de 4 núcleos a 1,7 GHz basado en la arquitectura ARM Cortex-A9. Cuenta con 2 GB de RAM unidos a la CPU.

La Ranura para tarjetas Micro Secure Digital (MicroSD).- Hay dos formas diferentes de almacenar el sistema operativo. Una es utilizar una tarjeta micro SD y la otra usar un módulo eMMC. Se utilizan normalmente como almacenamiento externo en los Smartphone y cámaras digitales.

La Ranura del Módulo eMMC.- El acceso al sistema de almacenamiento eMMC es 3-4 veces más rápido que la tarjeta SD. Hay 3 opciones de 8GB, 16GB y 64GB. El

uso de eMMC aumentará la velocidad y capacidad de respuesta, similar a la que se experimenta cuando se cambia un disco duro mecánico (HDD) por una Unidad de Estado Solido (SSD) en un PC.

La clavija de Energía.- Utilizado para la entrada de corriente de 5 voltios, con un diámetro interior de 0,8 mm y un diámetro exterior de 2,5 mm. El ODROID-U3 consume menos de 1A en la mayoría de los casos, pero puede subir a 2A si se le conecta directamente varios periféricos USB sin alimentación externa.

Los Puertos USB.- Hay 3 puertos USB host 2.0. Se puede conectar un teclado, ratón, adaptador WiFi entre otros dispositivos. También se puede cargar un Smartphone con estos puertos.

El botón de Corriente.- Se puede encender y apagar ODROID-U3. Una vez que se pulsa para apagar el sistema, se verá un cuadro de diálogo en pantalla para confirmar la petición de apagado.

Conector HDMI.- Para reducir al mínimo el tamaño de la placa se ha empleado un micro HDMI Tipo D. La salida HDMI soporta resoluciones de 720p y 1080p.

Puerto Ethernet.- El puerto Ethernet RJ45 estándar para conexiones LAN con una velocidad de 10/100 Mbps. La conexión Wi-Fi a través de un adaptador USB es otra opción que se puede utilizar.

LEDs indicadores de estados.- ODROID-U3 tiene cuatro indicadores LED que ofrecen información visual del estado operativo de la placa.

Audio Analógico.- Es un conector de audio analógico estándar de 3,5 mm. El conector tiene 4 polos para soportar una entrada de micrófono mono mediante los auriculares de los Smartphone Android.

Conector Micro USB.- Es el único conector Micro USB del dispositivo. Se puede usar con los drivers de Linux Gadget en un PC, lo que significa que los recursos de

ODROID-U3 se pueden compartir con los típicos PCs. No se puede utilizar como sistema de alimentación.

General Purpose Input and Output (GPIO).- Se puede usar para IRQ/I2C/UART en electrónica y robótica. Estas conexiones pueden ampliarse a muchos más puertos PWM/ADC/GPIO.

Puerto Terminal Serial.- Conectado a un PC permite el acceso a la consola de Linux. Se puede ver el registro del arranque o cambiar la configuración de vídeo o de red.

Salida PWM para el ventilador de refrigeración.- Cuando se trabaja con ODROID U3 en una habitación con una temperatura elevada, es útil conectar un pequeño ventilador. La salida PWM aumenta o disminuye la velocidad del ventilador de forma proporcional a la temperatura de la CPU.

RTC (Real time clock).- Conector batería de reserva. Si se desea añadir una función RTC para los archivos logs o el mantenimiento cuando se desconecta, basta con conectar una batería de reserva. RTC está implementado en ODROID-U3.

2.1.9.3 Los Periféricos

Ahora que se conoce la distribución de la placa, se necesita también conocer algunas cosas sobre los periféricos más apropiados (algunos se muestran más abajo) que puede ser utilizados con ODROID-U3.

Además de estos periféricos, también se necesita un monitor, un teclado y un ratón. Cualquier teclado y ratón inalámbricos o conectados por cable USB son compatibles con ODROID-U3. El monitor o TV debe soportar resoluciones de 720p (1280x720) o 1080p (1920x1080), aproximadamente.

Una fuente de Alimentación. Este es el periférico más importante. Se debe utilizar un adaptador de 5V/2A. El enchufe debe tener un diámetro interior de 0,8 mm y diámetro exterior de 2,5 mm, con centro positivo y exterior negativo.



Figura 2.15 Fuente de alimentación Odroid U3

Fuente: Revista Odroid Magazine Volumen 1 Pág. 6 (2014)

Una Tarjeta MicroSD. Si no se usa un módulo eMMC, se necesita una tarjeta MicroSD. Se recomienda una MicroSD de 8GB clase 10. Las tarjetas de clase más baja pueden afectar al rendimiento.



Figura 2.16 Micro SD Odroid U3

Fuente: Revista Odroid Magazine Volumen 1 Pág. 6 (2014)

Un Módulo eMMC. Si se quiere mejorar el rendimiento, debe escogerse el módulo eMMC. Es mucho más rápido que la MicroSD y también tiene preinstalado Ubuntu.

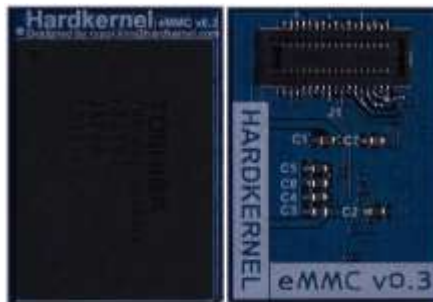


Figura 2.17 Modulo eMMC

Fuente: Revista Odroid Magazine Volumen 1 Pág. 6 (2014)

Un cable HDMI a micro HDMI. Se necesita este cable para conectar el U3 a un monitor. También podría utilizarse un adaptador para un monitor VGA ya que ODROID-U3 no tiene salida VGA. Se recomienda el cable HDMI de Hardkernel.



Figura 2.18 Cable HDMI a micro HDMI

Fuente: Revista Odroid Magazine Volumen 1 Pág. 6 (2014)

Adaptador WiFi. Se puede utilizar un adaptador USB WiFi en lugar de la conexión de red por cable. Se recomienda el adaptador WiFi RTL8188CUS.



Figura 2.19 Adaptador Wifi

Fuente: Revista Odroid Magazine Volumen 1 Pág. 6 (2014)

Caja. Para proteger la ODROID-U3 de daños inesperados. Se recomienda una carcasa protectora.



Figura 2.20 Case Odroid U3

Fuente: Revista Odroid Magazine Volumen 1 Pág. 6 (2014)

2.1.9.4 Arquitectura

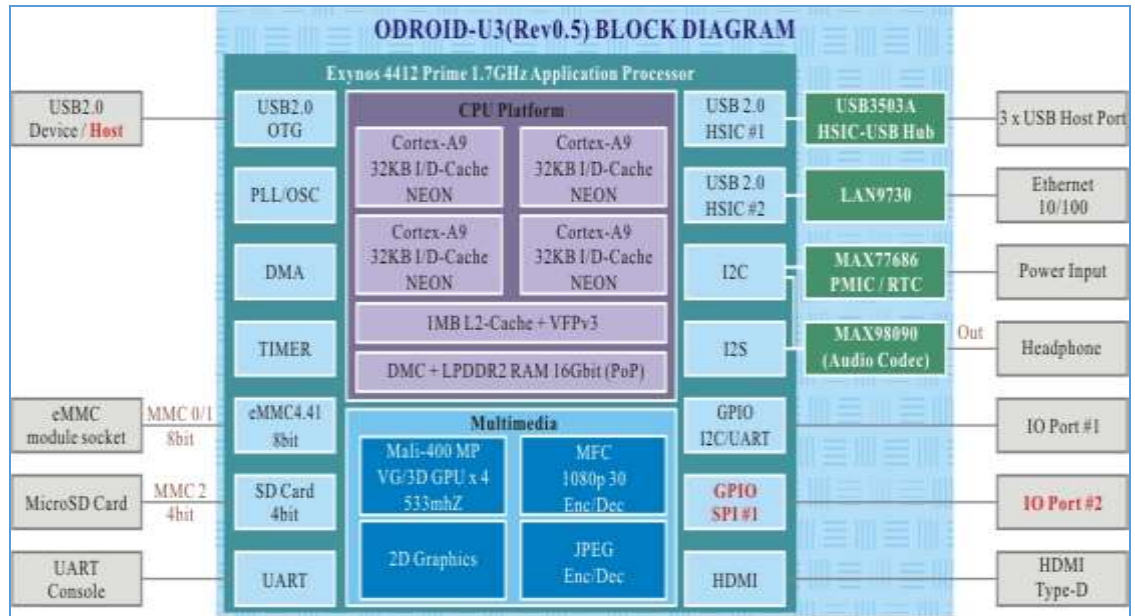


Figura 2.21 Arquitectura Odroid U3

Fuente: Revista Odroid Magazine Volumen 1 Pág. 6 (2014)

2.1.10 ¿Por qué ARM?

La arquitectura ARMv7 ha resultado ser un auténtico desafío para HPC (computación de alto rendimiento) en numerosas ocasiones en las que anteriores arquitecturas ARM no lo han sido. Al margen de su eficiencia energética, ARM se presenta como la tecnología en vía de desarrollo de los denominados “sistemas embebidos”. Cuando un centro de datos se compone de más de 10 mil núcleos, las mejoras graduales se traducen en importantes ventajas.

Estos pequeños cambios pueden representar un significativo ahorro de espacio y energía. Cuando la memoria está compartida entre la CPU y la GPU los SoCs ARM (Sistema en un Chip) duplican las extensiones SIMD (Una Instrucción, Múltiples Datos) en Cortex-A15 NEÓN GPU mejorando considerablemente el acceso a la memoria. Los beneficios aparecen en el momento adecuado, cuando la solicitud (aplicación) y el conjunto de datos se encuentran en la cache on-die.

Con la creciente acogida del sistema GP-GPU (General Purpose GPU) y la expansión de aplicaciones de tipo HPC basadas en aplicaciones complejas, los rápidos modelos de cálculo ARM son ahora más importantes que nunca, una tecnología con una participación cada vez mayor de HPC .

2.1.11 ¿Por qué ODROID?

Actualmente, ODROID tiene un procesador Exynos con 4 núcleos ARM. La próxima serie Exynos5 tiene 8 núcleos, 4 de los cuales son ARM Cortex-A15. Hardkernel así como el consorcio de investigación de Soluciones Informáticas han experimentado mejoras significativas de potencia y rendimiento en la serie XU, en comparación con otras arquitecturas de hoy en día. Con su ambicioso calendario de lanzamientos en nuevas tecnologías, Hardkernel ha adoptado esta plataforma siguiendo la ley de Moore y la tecnología ARM, lo que les ha permitido unirse a la ola de los más novedosos, mejores y de bajo coste sistemas de alto rendimiento, a la vez que su compatibilidad ha ido creciendo significativamente.

La tecnología ODROID, basada en ARM multinúcleo y una Unidad de procesamiento gráfico de aplicación general (GP-GPU), ofrece una tecnología micro servidor que está a punto de transformar el mundo de la informática en la nube, la informática paralela y los sistemas embebidos.

2.1.12 Tecnología big.LITTLE

El ODROID XU + E presenta arquitectura ARM big.LITTLE, con procesadores de bajo consumo y alta potencia: Cortex- A7 y Cortex- A15 con ocho núcleos.

¿Cómo se distribuyen los procesos?

El objetivo de la arquitectura big. LITTLE es reducir el consumo de energía usando los núcleos A15 más potentes cuando sea necesario. Los algoritmos de programación de procesos del kernel reflejan precisamente esto. Basado en los datos que se recopilaban usando sensores en el XU + E, se concluye que el algoritmo de programación de los procesos de migración del clúster funciona. En este algoritmo

de programación los procesadores A7 son usados hasta que la carga del procesador alcanza un cierto nivel, momento en el que los núcleos A15 se activan y se hacen cargo de cálculo hasta que la carga disminuye de nuevo.

Otros algoritmos de programación big.LITTLE incluyen In-Kernel Switcher (IKS) que empareja cada núcleo A7 con un núcleo A15 para la migración del proceso y Global Task Schedule (GTS), que distribuye los procesos entre los núcleos en base a sus necesidades de procesamiento. Esto hace que el bajo consumo y el alto rendimiento sean posibles en una única arquitectura. El procesador A15 se complementa con una GP-GPU que permite en una única instrucción multiplicar la aceleración de datos, además del procesamiento vectorial de precisión individual con las extensiones de coma flotante A15 NEON. Para muchas aplicaciones en la nube, la precisión individual es más que suficiente para los cálculos de coma flotante.

Aplicaciones en la nube

La informática en la nube basada en varios ODROID XU + E de baja potencia ofrece una interesante alternativa a procesadores x86-64 sumamente virtualizados. Al ofrecer muchos más sistemas pequeños, junto con una red gigabyte habilitada por el USB 3.0 del XU + E, las arquitecturas en la nube posibilitan los niveles de red al igual que las arquitecturas “más tradicionales” que virtualizan mediante la capa del sistema operativo host. Debido a que los sistemas son relativamente baratos, es posible un mayor control del hardware físico dentro de la infra-estructura en la nube. Esto es comparable a algunos de los proyectos actualmente, como HP Moonshot, hecho realidad hoy en día, sin marca registrada. Cualquiera puede desarrollar una nube XU + E.

Además, la capacidad E/S de los módulos eMMC ofrecen un rendimiento/velocidad superior a 100Mbyte/s y su capacidad llega actualmente hasta a los 64 gigabytes. Esto permite una mayor carga de trabajo.

Soporte para librerías matemáticas de alto rendimiento

Las librerías matemáticas tanto gratuitas como comerciales tienen o pronto tendrán soporte para ODROID. ATLAS, por ejemplo, (<http://source-forge.net/projects/math->

atlas) se encuentra en pleno desarrollo para la familia ARM y ya ha demostrado beneficios para muchas aplicaciones que utilizan procesos BLAS.

Además, las librerías comerciales para el procesamiento de imágenes y señales están en desarrollo, en particular las basadas en el estándar VSIPL (<http://www.omg.org/hot-topics/vsipl.htm>). Por ejemplo, “RunTime Computing Solutions” (www.runtime-computing.com) pronto lanzará VSIPL para varios núcleos ARM. Esto facilita la transición de plataformas más antiguas, como la familia de procesadores PowerPC Altivec directamente a ARM con el correspondiente traslado de rendimiento, logrando que éste sea más alto en las bibliotecas de código abierto como FFTW.

2.1.13 Problemas ODROID

Desafortunadamente, no existe el sistema perfecto y ODROID XU + E no iba a ser una excepción. Se ha tenido que solventar dos problemas cuando se desarrolla el clúster con NFS y LDAP por primera vez. Se encuentra que el kernel usado en la imagen de Xubuntu 13.10 en realidad no soportaba NFS. Este se solventó fácilmente con UNFS3 disponible gratuitamente y ejecutándose en el nodo principal.

En segundo lugar, se encuentra con algunos problemas al configurar los nodos para la autenticación en el servidor LDAP en el nodo principal usando nscd y ldap-auth-client. La solución a este problema pasaba por vincular simbólicamente el archivo `/lib/arm-linux-gnueabi/nss_ldap.so` a `/lib/libnss_ldap.so.2`. Las buenas noticias son que el soporte de Linux está evolucionando rápidamente, y se ve grandes mejoras en el rendimiento y la funcionalidad con cada versión.

2.1.13.1 Características Técnicas

Tabla 2.3 Tabla de Características

UPC	1.7GHz <u>Exynos4412</u> primer procesador Cortex-A9
-----	--

	de cuatro núcleos con PoP (paquete en paquete) 2GByte LPDDR2 880Mega Data Rate
PMIC	<u>MAX77686</u> Power Management IC de MAXIM
HSIC USB 2.0 Hub	<u>USB3503A</u> USB integrado concentrador compatible con 2.0 / HSIC puerto corriente arriba del SMSC / Microchip
Controlador HSIC Ethernet	<u>LAN9730</u> HSIC controlador USB 2.0 a Ethernet 10/100 con HP Auto-MDIX de SMSC / Microchip
Audio CODEC	<u>MAX98090</u> es un CODEC de audio de rendimiento completo y alta de MAXIM
Protección IC	<u>NCP372</u> Sobretensión, sobre intensidad de corriente, tensiones en sentido inverso la protección del IC de OnSemi.
Interruptor de carga USB	<u>NCP380</u> Protección IC para la fuente de alimentación USB de OnSemi.
HDMI acondicionador	<u>IP4791CZ12</u> HDMI protector interfaz transmisor con cambiador de nivel de NXP
Conector HDMI	Estándar Micro-HDMI, soporta hasta 1920 x 1080 de resolución
Conectividad	USB Host x 3, dispositivo x 1, Ethernet RJ-45, auricular Gato
IO Puertos	GPIO, UART, I2C, SPI (Junta de Revisión 0.5 o superior)
Ranura de almacenamiento	Ranura Micro-SD, conector del módulo eMMC
Entrada DC	5V de entrada / 2A, la especificación Plug es

diámetro interno de 0,8 mm y de 2,5 mm de diámetro exterior

Nota: Hardkernel (2013) Fuente:

http://www.hardkernel.com/main/products/prdt_info.php?g_code=G138745696275

2.1.14 ODROID-U3 VS RASPBERRY PI

Ambos son ordenadores compatibles con Linux, económicos y de una única placa ARM para diversos fines y propósitos.

Aunque ODROID-U3 es un económico ordenador de una sola placa ARM, incorpora un quad-core Samsung ARM. Sus especificaciones incluyen un SoC Samsung Exynos 4412 quad-core ARM Cortex-A9 a 1.7GHz, Mali-400 MP con 4 núcleos de GPU, tres puertos 2.0 host USB, un puerto Ethernet 10/100MB, salida de vídeo 1080p a través de micro -HDMI, 2 GB de memoria LP-DDR2, eMMC y micro-SD para el almacenamiento. Soporta los sistemas operativos Debian Wheezy, Ubuntu 13.10 Linux y Android Jellybean.

2.1.14.1 Hardware

Comparándolo con el Raspberry Pi, el número de núcleos de CPU/GPU es 4 veces más alto y la frecuencia de reloj de funcionamiento de la CPU es alrededor de 2,4 veces más rápida. Además, el tamaño de la RAM es también 4 veces mayor y la frecuencia acceso a la ésta es 1,6 veces más rápida. En superficie, el RPI es 1,2 veces más grande y el peso es muy similar si se considera el disipador de calor. También se puede utilizar en ODROID-U3 hardware RTC (reloj en tiempo real) con una pila tipo botón.

El RPI tipo B utiliza un SoC (System on a Chip) Broadcom que es un ARM v6 con una velocidad de reloj de 700Mhz (se puede incrementar hasta un 1GHz), sin embargo el RPI probado no superó los 800Mhz). En cambio, el U3 utiliza un

procesador ARM v7 más reciente, el Samsung Exynos 4412. Es el mismo chip que encontrarás en los Smartphone como el Galaxy Note 2 y el Galaxy S3. Dispone de 4 núcleos Cortex-A9 a 1.7Ghz (se puede incrementar hasta un 2.0Ghz). Recuerde que el “overclocking” (aumento de velocidad) es una cuestión de suerte. No todas las placas tendrá el mismo resultado “overclocking”.

Al igual que el RPi, el U3 tiene un puerto Ethernet 10/100. Aun así, el U3 muestra velocidades de transmisión más altas, ya que cuenta con un nodo separado. El ODROID-U3 incorpora 3 puertos USB permitiendo conectar más periféricos. Los puertos USB de U3 pueden proporcionar hasta 500 mA. El sistema no se reinicia al conectar dispositivos USB.

El U3, a diferencia de RPi, utiliza tarjetas microSD en lugar de una de tamaño normal. Las tarjetas microSD son cada vez más popular debido a que la mayoría de los Smartphone del mercado que permiten expandir su almacenamiento usan este tipo de tarjetas. Además, el U3 va aún más lejos y añade la opción de utilizar memoria eMMC. eMMC es una memoria flash MLC de alta velocidad como la usada como memoria interna en los teléfonos modernos y tabletas (eMMC se vende por separado como un accesorio). La velocidad de acceso a eMMC es 3-4 veces mayor que en las tarjetas SD.

El número de pines GPIO de E/S es menor que en RPI. Sin embargo, puedes ampliar E/S con la placa de expansión I2C E/S o el kit USB E/S, además de Arduino o interfaces similares.



Figura 2.22 Case Odroid U3

Fuente: Revista Odroid Magazine Volumen 1 Pág. 23 (2014)

ODROID-U3	Raspberry Pi (Modelo B / 512MB)
CPU SAMSUNG EXYNOS-4412 Prime 4 x ARM Cortex-A9 @1.7Ghz Arquitectura ARMv7	BROADCOM BCM2835 1 x ARMII @700Mhz Arquitectura ARMv6
GPU 4 x ARM Mali400 @400Mhz	1 x VideoCore IV @ 250 MHz
RAM 2GB LP-DDR2 @440Mhz	512MB SDRAM @400Mhz
USB 2.0 Host + device 3 Puertos • 1 Puerto Linux USB Gadget driver	2 Puertos • NO
ETHERNET 10/100 Mbit/s	10/100 Mbit/s
SALIDA VIDEO HDMI (480p/720p/1080p)	HDMI / Composite RCA
SALIDA AUDIO 3.5mm Jaok / HDMI	3.5mm Jaok / HDMI
RELOJ EN TIEMPO REAL SI (Conexión batería reserva)	NO (A menos que uses un módulo GPIO)
GPIO 5 (1.8Volt)	17 (3.3Volt)
TAMAÑO 83 x 48mm (3.27" x 1.89")	85.6 x 56mm (3.37" x 2.2")
PESO 30g (1.06 oz) 48g (1.69 oz) with a heat sink	45g (1.6 oz)
PRECIO 59\$	35\$

Figura 2.23 Comparación entre ODROID & RASPBERRY
Fuente: Revista Odroid Magazine Volumen 1 Pág. 23 (2014)

2.1.14.2 Rendimiento

Las pruebas se realizaron con las imágenes ofrecidas por el fabricante y con una instalación limpia, además de apt-get update && apt-get upgrade para que ambas placas estuviesen actualizadas.

RPI fue elevado a 800MHz y se usa una tarjeta SD Sandisk Extreme UHS-1 de 8GB con Debian Wheezy.

U3 fue ejecutado a su velocidad estándar de 1.7Ghz usando una eMMC de 16 GB con Ubuntu 13.10.

Ambas unidades fueron alimentadas por una fuente de 5V/2A y conectadas a la salida de HDMI 1920x1080.

Ejecutar el banco de prueba: Unix-Bench version 5.1.3

Activar los cuatro núcleos en ODROID-U3 con el comando. /Run-c 4.

La evaluación Dhrystone-2 es aproximadamente 14 veces más rápida en ODROID-U3. La evaluación E/S en torno a 5 veces más rápida en ODROID-U3, debido a que el sistema eMMC es más rápido. En términos globales, los resultados de las pruebas muestran que ODROID-U3 es alrededor de 8 veces más rápido que RPI. Sin embargo, el precio de la U3 es sólo 1,7 veces mayor que el del RPI.

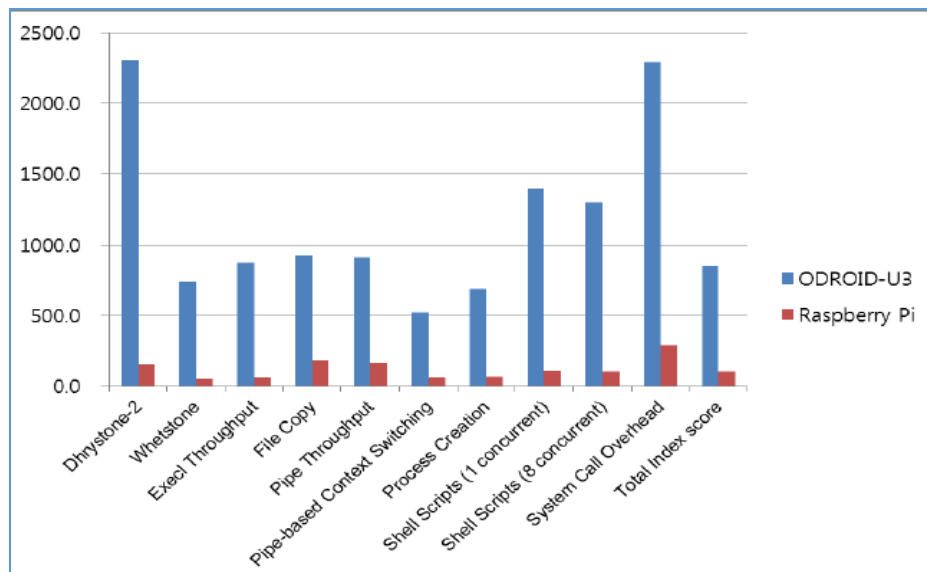


Figura 2.24 Comparación de Rendimiento

Fuente: Revista Odroid Magazine Volumen 1 Pág. 23 (2014)

2.1.14.3 Programación y Desarrollo

También se compara varios IDEs y compiladores para desarrollo de software. Se prueba Scratch, Arduino IDE, Python, compilación del Kernel Linux y Eclipse.

El U3 es, obviamente mucho más rápido al cargar el IDE y compilar el código, mientras que RPi no podía ejecutar Eclipse debido a la RAM insuficiente.

La compilación completa del código del kernel de Linux es 17 veces más rápida que en RPI.

Se utiliza la opción -j5 en el comando “make” para usar los 4 núcleos.

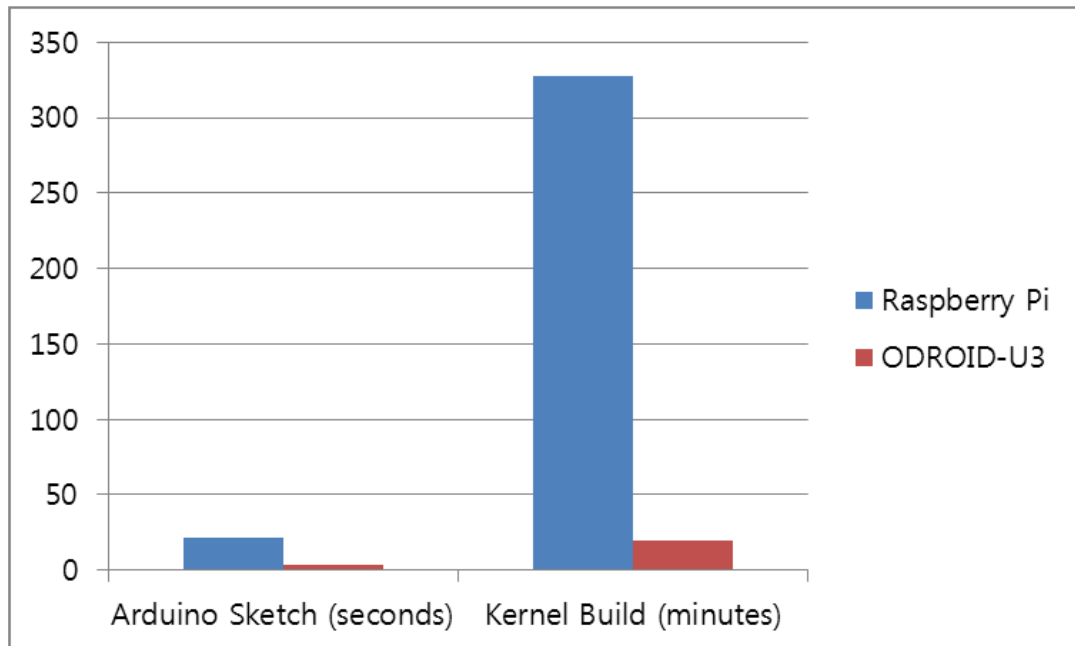


Figura 2.25 Comparación de Rendimiento

Fuente: Revista Odroid Magazine Volumen 1 Pág. 23 (2014)

Esquema Arduino: Tiempo de desarrollo de un simple esquema LCD

- RPi: 21.7 segundos
- U3: 3.2 segundos

Compilar Kernel: Tiempo de desarrollo del código fuente completo del kernel 3.4 de Linux

- RPi: 327 minutos (5 horas y 27 minutos)
- U3: 19 minutos

2.1.14.4 Conclusión

Como muestra los resultados anteriores, ODROID-U3 quad-core de 1,7 GHz supera claramente a Raspberry Pi con su único núcleo de 700MHz (incluso cuando se incrementa su velocidad). Muchos de los resultados de las pruebas muestran de seis a doce veces mejor rendimiento de la plataforma U3, siendo el aumento del coste de tan sólo 1,7 veces. Aunque ambas plataformas son dispositivos compatibles con linux, la relación rendimiento/coste es completamente diferente.

Si se considera adquirir un diminuto ordenador para uso general, desarrollar software o como plataforma de proyectos, ODROID-U3 ofrece mucha más satisfacción y diversión con un increíble rendimiento a un precio muy bajo.

2.2 SOFTWARE

2.2.2 ¿Qué es Qt Creator?

Qt Creator es un Entorno Integrado de Desarrollo o IDE (editor + compilador + depurador) bastante completo, moderno, potente, fácil de manejar, eficiente, abierto y gratuito, que permite el desarrollo rápido de aplicaciones en entornos MS Windows, Mac OS y Linux.

Algunos ejemplos de programas creados con las librerías Qt son Adobe Photoshop Álbum, Google Earth, KDE, Opera, Skype, VLC media player, etc.



Figura 2.26 Entorno Qt Creator

Fuente: Gstatic (2013). Recuperado de https://encrypted-bn1.gstatic.com/images?q=tbn:ANd9GcSg6mShIhhDHTVVyRMOWwg9sWn1yw_

2.2.1.1 ¿Por qué QT?

En realidad, como todo software que se precie, Qt Creator no es solo un conjunto de clases empaquetadas y listas para enlazar con ellas en tiempo de compilación, sino que Qt proporciona una serie de herramientas y documentación imprescindibles para el desarrollador:

- Qt Designer, para crear diálogos e interfaces gráficamente.
- Qt Linguist, una herramienta para crear aplicaciones multilingües.
- Qt Assistant, un generador de documentación.
- Qmake, un generador de Makefiles multiplataforma.

Las aplicaciones más populares que usan Qt son por supuesto las de KDE. Las librerías de KDE extienden y complementan a las de Qt, así que las aplicaciones KDE dependen de Qt. Hay otras aplicaciones interesantes hechas con Qt, pero que no son KDE. Las más populares son Scribus, Skype y muchas otras.

2.2.1.2 Características fundamentales de Qt Creator:

- Utiliza el lenguaje de programación orientado a objetos C++.
- Se basa en Qt, una librería multiplataforma y gratuita para la creación de interfaces gráficas, programación web, multihilo, bases de datos, etc.
- Características avanzadas de IDE: sintaxis coloreada, completación automática de código, ayuda sensible al contexto, inspector de objetos, diseñador visual, compilador y depurador integrado, etc.
- Permite realizar programación visual y programación dirigida por eventos.

Programación visual: el programador centra su atención en diseñar el aspecto gráfico de la aplicación, la distribución de los elementos visuales (llamados widgets: formularios, botones, menús, cuadros de texto, etc.), la interacción entre los mismos, los distintos tipos de ventanas existentes, etc.

- Un entorno de programación visual se asemeja a un programa de dibujo, donde la imagen es una ventana (o formulario), y los elementos para dibujar son botones, etiquetas de texto, menús, etc.
- El programador diseña el aspecto gráfico que tendrá la aplicación.

Programación dirigida por eventos: el programador escribe el código que se ejecutará en respuesta a determinados eventos (llamados slots: pulsar un botón, elegir una opción del menú, abrir o cerrar una ventana, etc.).

- No existe la idea de un control de flujo secuencial en el programa, sino que el programador toma el control cuando se dispara un evento.
- La labor del programador es asociar a cada evento el comportamiento adecuado.

Las ventanas son clases, los componentes (widgets) son clases, y los eventos (slots) son métodos de las ventanas. Nuestra ventana es una subclase de la clase ventana (QMainWindow, QDialog o QWidget).

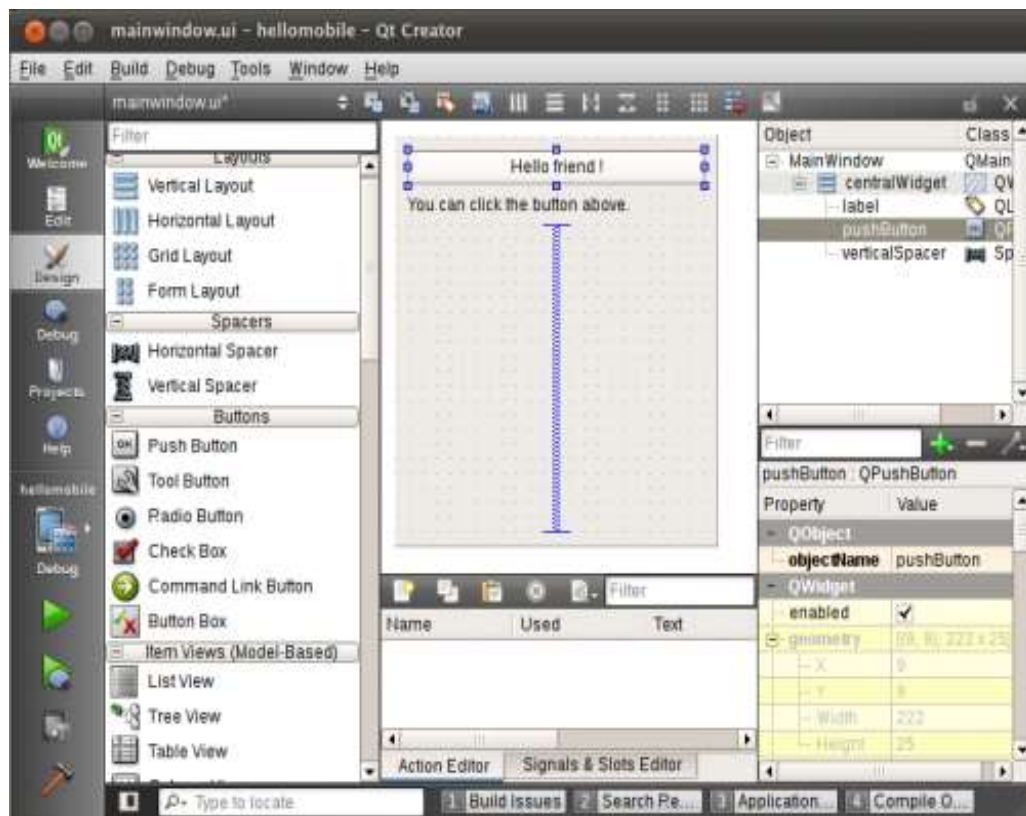


Figura 2.27 Qt creator

Fuente: Nokia (2015). Recuperado de

<http://developer.nokia.com/community/wiki/images/thumb/5/51/Qt-creator-new-mobile-qt-project-6.png/500px-Qt-creator-new-mobile-qt-project-6.png>

El principal motivo por el que Qt se ha hecho tan popular es por su propiedad de ser multiplataforma, esto es, con un mismo código, las aplicaciones se pueden compilar

sin tocar una sola línea en distintos sistemas operativos. Esto es posible porque los desarrolladores de Qt se han preocupado de generar diversas versiones de la librería: Qt/Windows (MS Windows 95/98/Me, NT4, 2000 y XP), Qt/X11 (Linux, *BSD, Solaris, HP-UX, IRIX, AIX, y otras variantes de UNIX que usen X11), y Qt/Mac (Apple Mac OS X). En definitiva, se programa una sola vez, y se compila contra la variante apropiada de Qt.

Las librerías Qt no sólo están disponibles para programar en C++, sino que hay soluciones para ser usadas con Python (PyQt), Java (Jambi), Javascript, etc.

2.2.1.3 ¿Cómo está disponible Qt?

Qt funciona con lo que se denomina licencia dual:

Licencia comercial: es la apropiada si se van a crear aplicaciones propietarias y no se va a distribuir y compartir el código.

Licencia Open Source: si la aplicación que se desarrolle va a hacer uso de alguna de las licencias compatibles GPL.

La versión que viene en las distribuciones de linux, es la Qt Open Source Edition, que está cubierta bajo la GPLv2 (y por tanto, con código fuente incluido), y que está disponible para X11, Windows y Mac. Esta edición incluye la última versión de Qt, y todas sus características. Por las características de la licencia GPLv2, si se opta por la Qt Open Source

Edition, nuestro código, al enlazarse contra ellas, deberá estar bajo una de las licencias libres válidas (GPL, LGPL, BSD, etc.).

En definitiva, Qt no exige dinero salvo que el desarrollador obtenga beneficio económico del uso de su librería, en cuyo caso es obligatorio y se podría decir que lógico y ético agradecer económicamente el esfuerzo que han realizado para proporcionar la librería.

La versión más reciente, a la fecha de publicación de este trabajo de titulación, disponible en formato Open Source es la 4.3.3.

2.2.1.4 ¿Cómo obtener e instalarse Qt?

En Linux es trivial su instalación, ya que se instalan con el entorno de ventanas KDE. Si aun así no esta instalada la librería, se la descarga de la web de Trolltech (la URL exacta para la versión linux es: <http://trolltech.com/developer/downloads/qt/x11> y seguir las instrucciones indicadas en el archivo: (<ftp://ftp.trolltech.com/qt/source/INSTALL>).

Se puede descargar las fuentes y compilar e instalar las librerías, como bajarse uno de los paquetes pre compilados preparados específicamente para su distribución de Linux (Red Hat, Suse, Debian, etc...).

Al ser trivial ésta última opción, la práctica No 8 de esta guía, detalla los pasos a seguir para una instalación desde las fuentes repositorios.

2.2.2 ¿Qué es Python?

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible.

Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos.



Figura 2.28 Logo python

Fuente: Citizen Scientists League (2014). Recuperado de <http://citizenscientistsleague.com/wp-content/uploads/2014/08/python.jpg>

2.2.2.1 ¿Por qué Python?

Python es un lenguaje que todo el mundo debería conocer. Su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo, muy rápido y, lo que es más importante, divertido. La sintaxis de Python es tan sencilla y cercana al lenguaje natural que los programas elaborados en Python parecen pseudocódigo. Por este motivo se trata además de uno de los mejores lenguajes para comenzar a programar.

Python no es adecuado sin embargo para la programación de bajo nivel o para aplicaciones en las que el rendimiento sea crítico. Algunos casos de éxito en el uso de Python son Google, Yahoo, la NASA, Industrias Light & Magic, y todas las distribuciones Linux, en las que Python cada vez representa un tanto por ciento mayor de los programas disponibles.

2.2.2.2 Características fundamentales de Python:

Lenguaje interpretado o de script

Un lenguaje interpretado o de script es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora (lenguajes compilados).

La ventaja de los lenguajes compilados es que su ejecución es más rápida. Sin embargo los lenguajes interpretados son más flexibles y más portables. Python tiene, no obstante, muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi interpretado. En Python, como en Java y muchos otros lenguajes, el código fuente se traduce a un pseudo código máquina intermedio

llamado bytecode la primera vez que se ejecuta, generando archivos “pyc o pyo”(bytecode optimizado), que son los que se ejecutarán en sucesivas ocasiones.

Tipado dinámico

La característica de tipado dinámico se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo del valor al que se asigne, y el tipo de esta variable puede cambiar si se le asigna un valor de otro tipo.

Fuertemente tipado

No se permite tratar a una variable como si fuera de un tipo distinto al que tiene, es necesario convertir de forma explícita dicha variable al nuevo tipo previamente.

Por ejemplo, si se tiene una variable que contiene un texto (variable de tipo cadena o string) no se puede tratar como un número (sumar la cadena “9” y el número 8). En otros lenguajes el tipo de la variable cambiaría para adaptarse al comportamiento esperado, aunque esto es más propenso a errores.

Multiplataforma

El intérprete de Python está disponible en multitud de plataformas (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS, etc.) por lo que si no se utiliza librerías específicas de cada plataforma nuestro programa podrá correr en todos estos sistemas sin grandes cambios.

Orientado a objetos

La orientación a objetos es un paradigma de programación en el que los conceptos del mundo real relevantes para nuestro problema se trasladan a clases y objetos en nuestro programa.

La ejecución del programa consiste en una serie de interacciones entre los objetos. Python también permite la programación imperativa, programación funcional y programación orientada a objetos.

```
#!/usr/bin/env python2.7
# by Alex Eames of http://RasPi.TV

import RPi.GPIO as GPIO
import sys, os
from time import sleep

GPIO.setmode(GPIO.BCM)                                # initialise RPi.GPIO
GPIO.setup(25, GPIO.OUT)                              # set up ports for output

answer = raw_input("Are you sure you want to reset? y/n \n")

if answer != "y" and answer != "yes":
    print "OK. Not resetting. To change your mind, rerun the program"
    GPIO.cleanup()
    sys.exit()

try:
    print "OK. Syncing file system, then resetting."
    command = os.system("sync")
    if command == 0:
        print "sync successful - resetting now"
        sleep(1)
        GPIO.output(25, 1)
        sleep(10)
        GPIO.output(25, 0)
        print "I think you forgot to connect your wires. connect them up and rerun the program."
        GPIO.cleanup()
except KeyboardInterrupt:                             # trap a CTRL+C keyboard interrupt
    GPIO.cleanup()
```

Figura 2.29 Programar Python

Fuente: RasPi.TV (2012). Recuperado de http://raspi.tv/wp-content/uploads/2012/12/reset.py_.gif

2.2.2.3 Instalación de Python

Existen varias implementaciones distintas de Python: CPython, Jython, IronPython, PyPy, etc. CPython es la más utilizada, la más rápida y la más madura. Cuando la gente habla de Python normalmente se refiere a esta implementación. En este caso tanto el intérprete como los módulos están escritos en C.

Jython es la implementación en Java de Python, mientras que IronPython es su contrapartida en C# (.NET). Su interés estriba en que utilizando estas implementaciones se pueden utilizar todas las librerías disponibles para los programadores de Java y .NET. PyPy, por último, como se puede suponer por el nombre, se trata de una implementación en Python de Python.

CPython está instalado por defecto en la mayor parte de las distribuciones Linux y en las últimas versiones de Mac OS. Para comprobar si está instalado se debe abrir una terminal y escribir python. Si está instalado se inicia la consola interactiva de Python y se obtiene texto parecido a lo siguiente:

```
Python 2.5.1 (r251:54863, May 2 2007, 16:56:35)
[GCC 4.1.2 (Ubuntu 4.1.2-0ubuntu4)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> Python para todos 10
```

La primera línea indica la versión de Python que se tiene instalada. Al final se puede ver el prompt (>>>) el cual indica que el intérprete está esperando código del usuario. Se puede salir escribiendo exit (), o pulsando Control + D.

Si no se muestra algo parecido no hay que preocuparse, instalar Python es muy sencillo. Se puede descargar la versión correspondiente a los principales sistemas operativos desde la web de Python, en <http://www.python.org/download/>. Existen instaladores para Windows y Mac OS. Si se utiliza Linux es muy probable que se pueda instalarlo usando la herramienta de gestión de paquetes de la distribución correspondiente, aunque también es posible descargar la aplicación compilada desde la web de Python.

2.2.3 ¿Qué es Scilab?

Scilab es un software matemático, con un lenguaje de programación de alto nivel, para cálculo científico, interactivo de libre uso y disponible en múltiples sistemas operativos (Mac OS X, GNU/Linux, Windows). Desarrollado por INRIA (Institut National de Recherche en Informatique et en Automatique) y la ENPC (École Nationale des Ponts et Chaussées) desde 1990, por Scilab Consortium dentro de la fundación Digiteo desde 2008, Scilab es ahora desarrollado por Scilab Enterprises desde julio 2012.

Scilab fue creado para hacer cálculos numéricos aunque también ofrece la posibilidad de hacer algunos cálculos simbólicos como derivadas de funciones

polinomiales y racionales. Posee cientos de funciones matemáticas y la posibilidad de integrar programas en los lenguajes más usados (Fortran, Java, C y C++). La integración puede ser de dos formas: por ejemplo, un programa en Fortran que utilice Scilab o viceversa. Scilab fue hecho para ser un sistema abierto donde el usuario pueda definir nuevos tipos de datos y operaciones entre los mismos.



Figura 2.30 Log Scilab

Fuente: INRIA Metalau Project / ENPC Cermics (2000). Recuperado de http://cermics.enpc.fr/scilab_new/images/scilab_logo.gif

Scilab viene con numerosas herramientas: gráficos 2-D y 3-D, animación, álgebra lineal, matrices dispersas, polinomios y funciones racionales, Simulación: programas de resolución de sistemas de ecuaciones diferenciales (explícitas e implícitas), Xcos: simulador por diagramas en bloque de sistemas dinámicos híbridos, Control clásico, robusto, optimización LMI, Optimización diferenciable y no diferenciable, Tratamiento de señales, Grafos y redes, Scilab paralelo empleando PVM, Estadísticas, Creación de GUIs, Interfaz con el cálculo simbólico (Maple, MuPAD), Interfaz con TCL/TK.

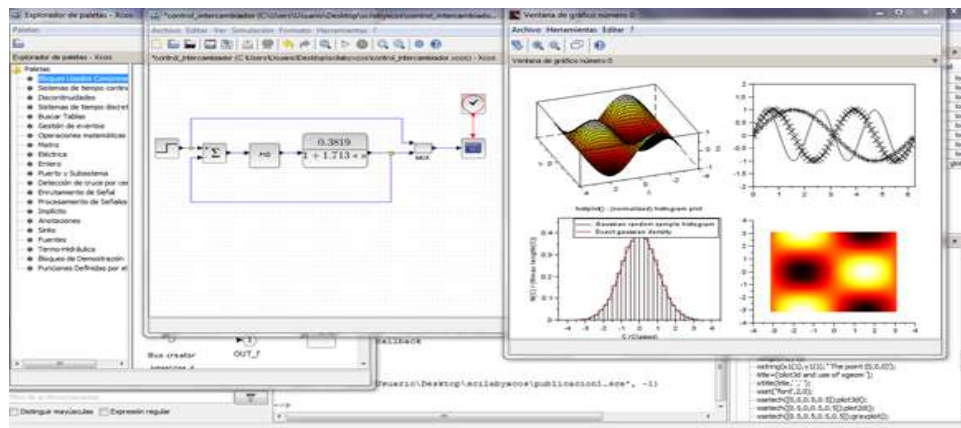


Figura 2.31 Entorno Scilab

Fuente: INRIA Metalau Project / ENPC Cermics (2000). Recuperado de http://cermics.enpc.fr/scilab_new/images/

Además se puede agregar numerosas herramientas o toolboxes hechas por los usuarios como Grocer una herramienta para Econometría u Open FEM (Una caja de Herramientas para Elementos Finitos), hecha por INRIA.

En el pasado Scilab podía ser utilizado en el análisis de sistemas, pero no podía interactuar con el exterior. Hoy en día se puede construir interfaces para que desde Scilab se pueda manejar un dispositivo, se conecte a la red a través de Tcp (Protocolo de Control de Transmisión) o Udp (User Datagram Protocol), etc.

Esto brinda la posibilidad de conectar una placa de adquisición de datos a Scilab y de esta forma el control de una planta on-line.

2.2.3.1 Xcos

El programa Scilab tiene un entorno similar a Simulink de Matlab para simulación de sistemas dinámicos y resolución de sistemas de ecuaciones diferenciales. Este entorno posee varios paquetes que incluye algunas herramientas para simulación sencilla de circuitos eléctricos y termo hidráulica.

Para lanzar el entorno Xcos desde Scilab basta con escribir el siguiente comando:

- xcos

Se lanza una nueva ventana, desde en la cual se insertan y conectan todos los bloques a voluntad del usuario.

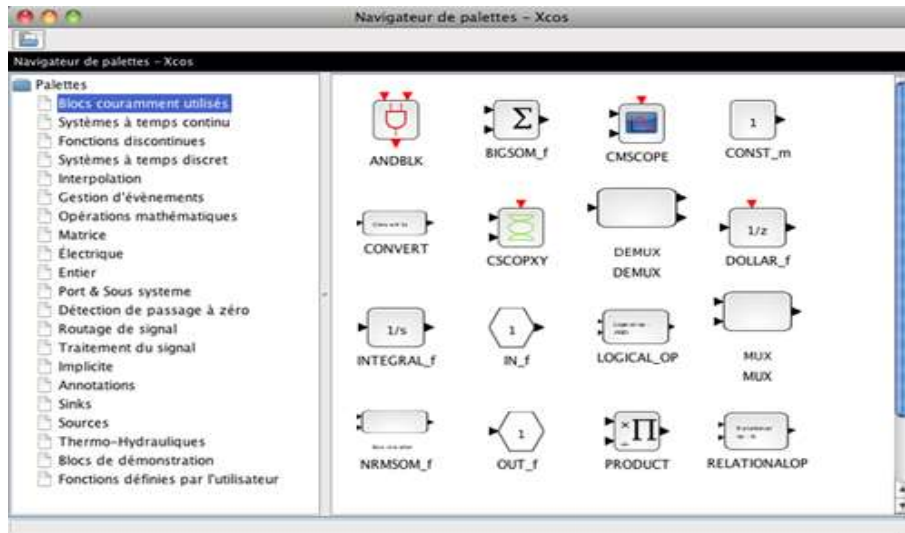


Figura 2.32 Funciones en Scilab

Fuente: Scilab.org (2015). Recuperado de https://www.scilab.org/var/ezflow_site/storage/images/media/images/gallery/xcos/palettes/3226-1-fre-FR/Palettes.png

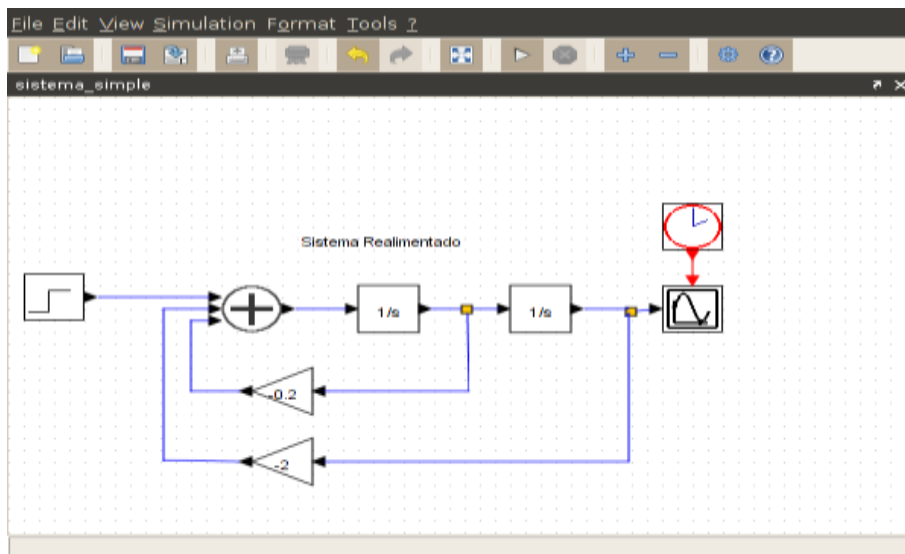


Figura 2.33 Navegando en Scilab

2.2.3.2 Paquetes disponibles para Scilab (Toolbox)

A partir de la versión 5.2 se puede consultar: <http://atoms.scilab.org/> para obtener un listado de módulos que extienden las capacidades de Scilab.

Para instalar módulos, se puede abrir Atoms haciendo click en Applications->Module Manager ATOMS, se selecciona la aplicación y ATOMS la descarga de internet y la instala.

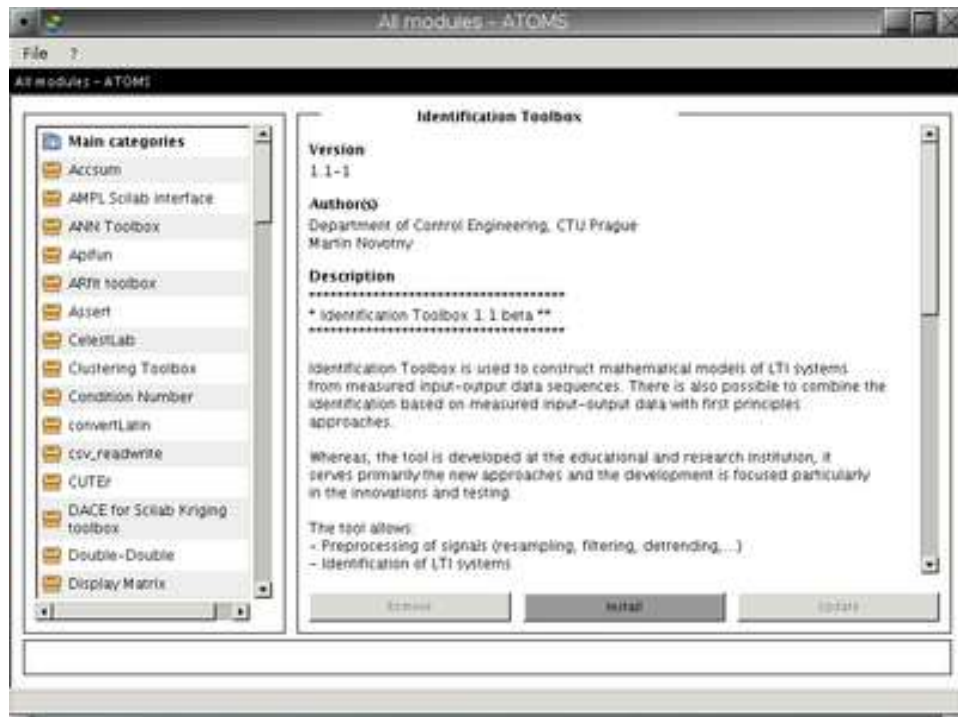


Figura 2.34 Módulos en Scilab

2.2.3.3 Instalación de Scilab.

Se recomienda, instalar la nueva versión del scilab que a la fecha de publicación de este trabajo es la 5.3.3.

Esta versión se puede bajar desde el sitio oficial de scilab que es:

- www.scilab.org

Cuando se entra se ve una ventana, en la que se ofrece bajar la versión de Windows y si se hace click en Other Systems, se encuentra también linux, mac etc. Si se quiere la versión en Windows simplemente se hace click sobre el recuadro que dice Download Scilab y debe comenzar la descarga de un archivo que tendrá un nombre parecido a éste: scilab-5.3.3.exe.

Cuando termina de descargar ese archivo, lo ejecuta (haciendo doble click sobre el nombre) y elegir la instalación “full”. En lugar de la instalación “full”, se puede seleccionar un subconjunto de “paquetes” a instalar. Durante el proceso de instalación les va a preguntar el idioma.

Una vez que instalaron esta versión de scilab cuando lo ejecuten se abrirá una ventana como la que se muestra a continuación.

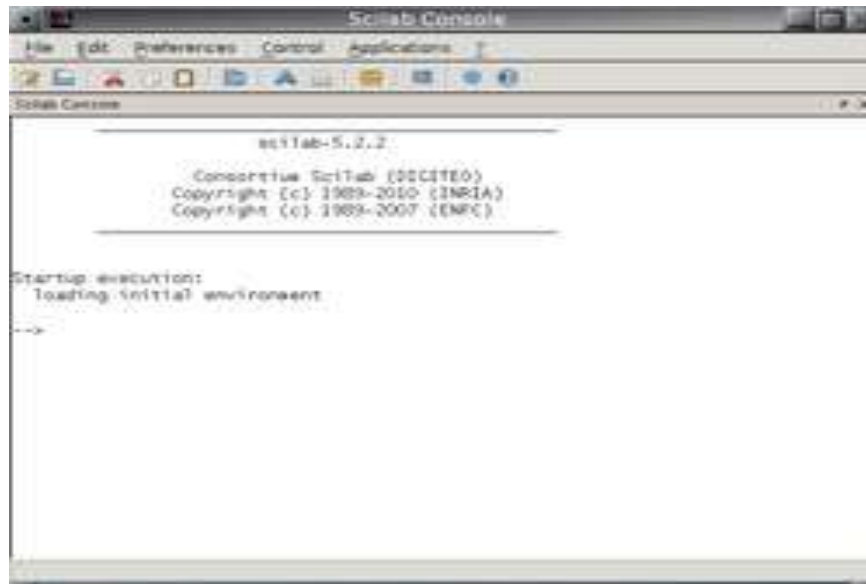


Figura 2.35 Instalación en Scilab

Fuente:

Una herramienta muy interesante que tiene esta nueva versión de scilab es el buscador e instalador de módulos adicionales (ATOMS). Los módulos adicionales son paquetes que no vienen con el programa (aún cuando se selecciona la instalación “full”) pero que uno puede agregar a medida que los necesita.

2.2.4 ¿Qué es GNU OCTAVE?

Octave o GNU Octave es un programa libre para realizar cálculos numéricos. Como su nombre indica, es parte del proyecto GNU. Es considerado el equivalente libre de MATLAB. Entre varias características que comparten, se puede destacar que ambos ofrecen un intérprete, permitiendo ejecutar órdenes en modo interactivo. Octave proporciona una interfaz de línea de comandos para resolver problemas lineales y no

lineales de manera numérica, y desarrollar otros experimentos numéricos utilizando para ello un lenguaje que en su mayoría es compatible con Matlab. También se puede utilizar como un lenguaje de lotes (batch-oriented language).

Octave tiene una gran cantidad de herramientas para resolver problemas de álgebra numérica comunes, encontrar las soluciones de ecuaciones no lineales, realizar integrales de funciones ordinarias, manipular polinomios, e integrar ecuaciones diferenciales ordinarias y ecuaciones diferenciales algebraicas. Es fácil de extender y modificar a través de funciones definidas por el usuario escritas en el propio lenguaje de Octave, o utilizando módulos cargados dinámicamente escritos en otros lenguajes como C, C++, Fortran, etc.

El proyecto fue creado alrededor del año 1988, pero con una finalidad diferente: ser utilizado en un curso de diseño de reactores químicos. Posteriormente, en el año 1992, se decidió extenderlo, y comenzó su desarrollo a cargo de John W. Eaton.¹ La primera versión alpha fue lanzada el 4 de enero de 1993. Un año más tarde, el 17 de febrero de 1994, apareció la versión 1.0.

El nombre surge de Octave Levenspiel, profesor de uno de los autores y conocido por sus buenas aproximaciones, por medio de cálculos mentales, a problemas numéricos en ingeniería química.

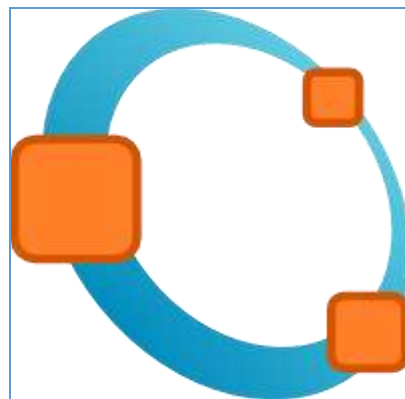


Figura 2.36 Logo GNU OCTAVE

Fuente: John W. Eaton (2013). Recuperado de <https://www.gnu.org/software/octave/images//logo.png>

2.2.4.1 Detalles técnicos:

- Octave está escrito en C++ usando la biblioteca STL.
- Tiene un intérprete de su propio lenguaje (de sintaxis casi idéntica a Matlab), y permite una ejecución interactiva o por lotes.
- Su lenguaje puede ser extendido con funciones y procedimientos, por medio de módulos dinámicos.
- Utiliza otros programas GNU para ofrecer al usuario crear gráficos para luego imprimirlos o guardarlos (Grace).
- Dentro del lenguaje también se comporta como una consola de órdenes (shell). Esto permite listar contenidos de directorios, por ejemplo.
- Además de correr en plataformas Unix también lo hace en Windows.
- Puede cargar archivos con funciones de Matlab (reconocibles por la extensión .m).

Tiene ayuda en español.

En el lenguaje Octave, la sintaxis es casi idéntica a la utilizada en MATLAB.

- Es un lenguaje interpretado.
- No permite pasar argumentos por referencia. Siempre son pasados por valor.
- No permite punteros.
- Se pueden generar scripts.
- Soporta gran parte de las funciones de la biblioteca estándar de C. Puede ser extendido para ofrecer compatibilidad con las llamadas al sistema UNIX.
- El lenguaje está pensado para trabajar con matrices, y provee mucha funcionalidad para trabajar con éstas.
- Soporta estructuras similares a los "struct"s de C.
- Al ser su licencia pública general de GNU, puede ser compartido y utilizado libremente.

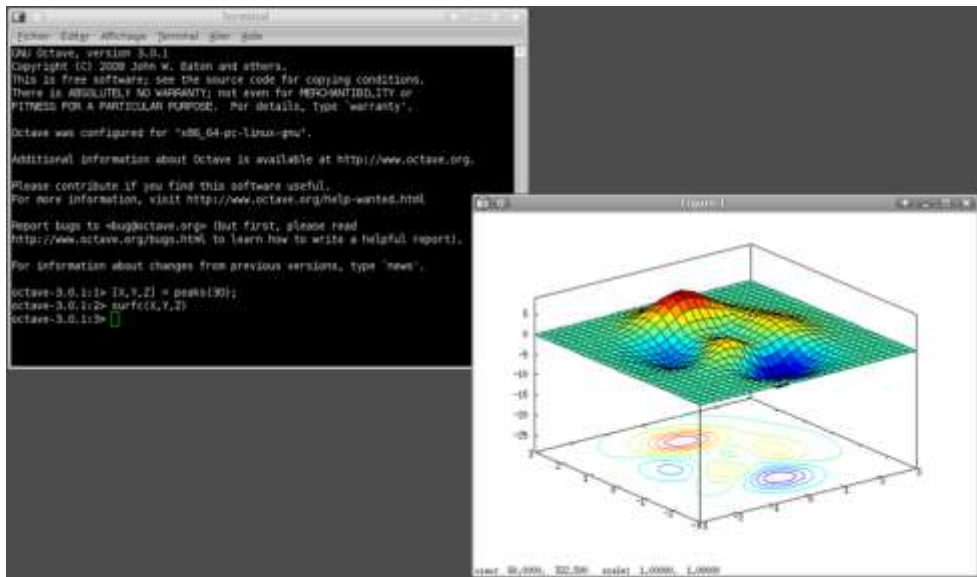


Figura 2.37 Entorno GNU OCTAVE

Fuente: IEEE (2015). Recuperado de

http://ewh.ieee.org/sb/el_salvador/uca/images/gnu-octave.png

2.2.4.2 Instalación en GNU/Linux

Para instalar Octave en GNU/Linux basta instalarlo desde los propios repositorios de su distribución, ya que GNU/Octave suele venir incorporado en las principales distribuciones. Para instalar GNU/Octave en una distribución que no lo incorpore, o si se prefiere instalar GNU/Octave directamente desde las fuentes, diríjirse a la web oficial y seguir las instrucciones de instalación

CAPÍTULO 3

ESTRUCTURA DEL DESARROLLO DE LAS PRÁCTICAS.

3.1 Estructura de los kits para el desarrollo de las prácticas

Raspberry-Pi y ODROID-U3 son herramientas excelentes para desarrollar habilidades como programador. Ideales para proyectos de control, domótica, automatización y robótica.

Debido a su versatilidad, bajo costo y gran documentación de soporte existente en la actualidad se han escogido estas placas en la implementación de los Kits didácticos de laboratorio para el desarrollo de aplicaciones con sistemas embebidos.

3.2 Componentes que lo conforman

Estos kits (Figura 3.1) estarán conformados por los siguientes elementos:

- Placa Raspberry –P modelo B (computador embebido)
- Placa ODROID-U3 (computador embebido).
- Memorias SD de 16 Gb.
- Hub USB de seis puertos
- Modem Wi-Fi TP-LINK TL-WN725N
- Modem Wi-Fi TP-LINK TL-WN722N
- Teclado genérico de 105 teclas
- Mouse genérico
- Cámara USB (web-cam)
- Placa interface para puerto GPIO (Raspberry- Pi)
- Placa interface para puerto GPIO U3 IO Shield (ODROID-U3)
- Adaptador de alimentación 5V 2 A.
- Convertidor de video HDMI a VGA
- Maleta contenedora de elementos del kit.

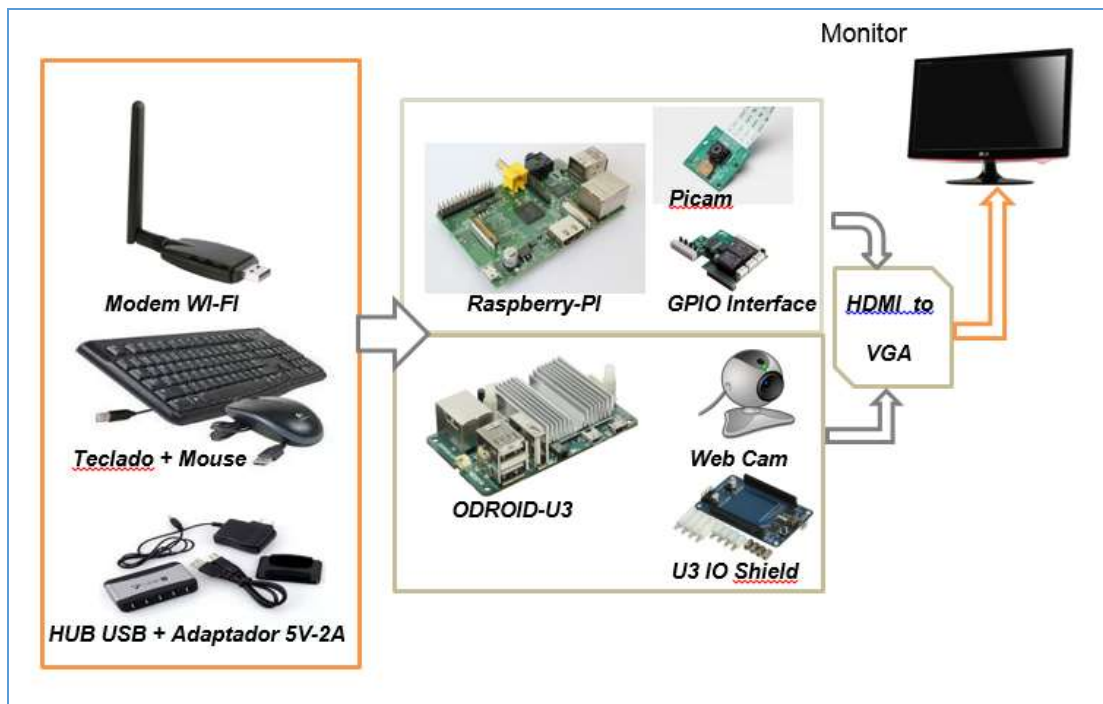


Figura 3.1 Equipos necesario dentro de maleta contenedora

Los componentes que conforman el Kit facilitan en gran manera la implementación de diversas prácticas, pues debido a su modularidad y dependiendo de la aplicación a realizar, se utilizan los componentes necesarios para ensamblar el conjunto que permite obtener los resultados del tema de estudio.

De manera general, las prácticas más significativas que se pueden realizar con estos kits son las siguientes:

- Configuración inicial e instalación de programas empleados en control electrónico.
- Configuración de la conexión a la red inalámbrica y sistemas de video vigilancia.
- Configuración y uso de los pines digitales del puerto GPIO.
- Configuración y uso de los pines UARTs para la comunicación RS-232 en el puerto GPIO.
- Introducción a PHYTON
- Control de un motor de pasos y de un servomotor programando en PHYTON y utilizando el C.I. ULN2803.

- Introducción a Octave, Scilab. Comparación entre MATLAB y Octave, Scilab: prestaciones, ventajas, alcances.
- Introducción a QT Creator y desarrollo de interfaces gráficas empleadas en instrumentación virtual y comparación con Lab-View.

3.3 Objetivos de su uso

Con el uso de estos Kits se logran los siguientes objetivos:

- Afianzar de manera práctica los conocimientos adquiridos en las diversas materias relacionadas a la electrónica analógica, digital, comunicaciones, control e instrumentación que se estudian a lo largo de la carrera de ingeniería electrónica.
- Conocer y aplicar técnicas de Programación en Python, QT, HTML y varios lenguajes de programación multiplataforma.
- Fomentar el desarrollo de proyectos con computadores Embebidos para afianzar y agilizar el aprendizaje de su estructura y funcionamiento.
- Fomentar y fortalecer el uso de software libre de licencias en el desarrollo de sistemas de monitoreo e instrumentación virtual.
- Adquirir y transmitir datos a través de los puertos de los sistemas embebidos para desarrollar aplicaciones de control.
- Establecer comparaciones entre las prestaciones y alcances existentes entre los programas MATLAB y Octave, Scilab.
- Establecer comparaciones entre las prestaciones y alcances existentes entre los programas Lab-View y QT Creator para aplicaciones de monitoreo e instrumentación industrial.
- Facilitar el desarrollo e implementación de proyectos de robótica y domótica.

CAPÍTULO 4

PRÁCTICAS Y RESULTADOS DEL PROYECTO

En el presente Capítulo 4 se ejecuta en diversas prácticas cada función de los sistemas embebidos, tanto de la Odroid-U3 como de la Raspberry Pi.

La idea es consolidar paso a paso la destreza para el manejo y aplicaciones de estas nuevas tecnologías.

En las tres primeras prácticas se conoce el manejo y funcionabilidad de estos sistemas. El estudiante se familiariza y afianza en el uso y operación, lo cual es la base para las restantes prácticas.

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

PRÁCTICA No 1

NÚMERO DE ESTUDIANTES: 20

ING. LUIS CÓRDOVA

TIEMPO ESTIMADO 2 HORAS

TEMA:

**CONFIGURACIÓN INICIAL E INSTALACIÓN DE PROGRAMAS
EMPLEADOS EN CONTROL ELECTRÓNICO.**

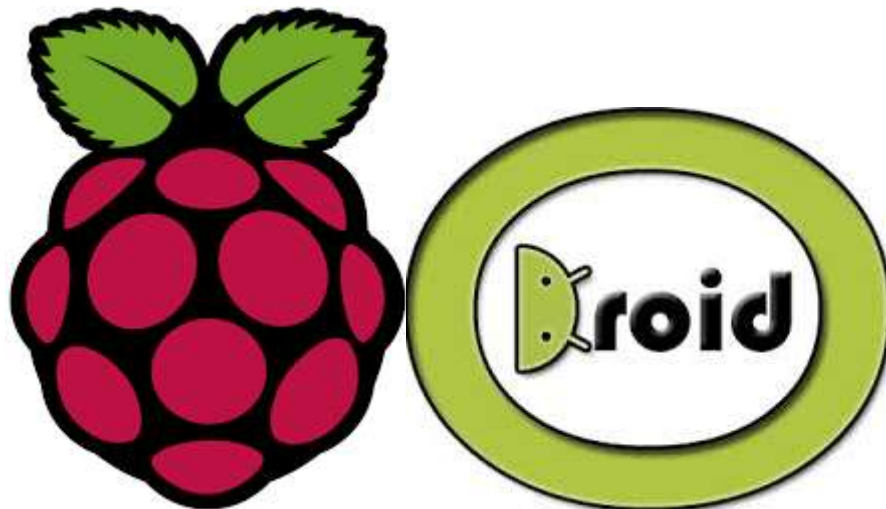


Figura 4.1 Logo de Equipos empleados

1. OBJETIVO GENERAL.

- Familiarizar al estudiante a las configuraciones y primeros pasos a seguir en el uso y manipulación de sistemas embebidos.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

2. OBJETIVOS ESPECÍFICOS.

- Como crear una SD de arranque para la Raspberry PI.
- Familiarizar al estudiante al sistema operativo Raspbian.
- Como utilizar el Win 32 Disk Imager.
- Como crear una SD de arranque para la Odroid U3.
- Familiarizar al estudiante al sistema operativo Lubuntu.

3. MARCO TEÓRICO.


¿Qué es Raspbian?

Raspbian es un sistema operativo Linux basado en Debian, optimizado para Raspberry. De entre todas las distribuciones Linux, Debian es la más conocida y popular, gracias a su estabilidad. Muchas otras distribuciones, como Ubuntu o Linux Mint están basadas en esta distribución.



Figura 4.2 Fusión y resultado de Raspbian

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Raspbian es la distribución más recomendada para Raspberry Pi, ya que además incluye unos 35.000 paquetes de software especialmente pre compilados para Raspberry.

Raspbian ofrece un sistema operativo completo Linux, con un entorno gráfico de escritorio basado en LXDE, que es muy ligero y funcional.

¿Qué es Lubuntu?

Lubuntu es una distribución oficial del proyecto Ubuntu que tiene por lema "menos recursos y más eficiencia energética", usando el gestor de escritorio LXDE. El nombre Lubuntu es una combinación entre LXDE y Ubuntu.

Es una distribución liviana (el nombre viene de "light" en inglés) y rápida de Ubuntu. Es decir, su objetivo es ofrecer las características que hacen a Ubuntu una de las mejores distribuciones pero reduciendo el uso de recursos en ordenadores con características más limitadas. Así como Ubuntu cuenta con un escritorio Unity (caracterizado por su importante consumo de recursos) o Gnome (no consume tanto como Unity pero no es lo más recomendado para ordenadores con pocos recursos), Lubuntu utiliza el escritorio minimalista LXDE (Lightweight Desktop Environment o Entorno de Escritorio Liviano).

El gestor LXDE usa el administrador de ventanas Openbox e intenta ser un sistema operativo que demande pocos recursos de RAM, CPU y otros componentes, especialmente ideados para equipos portátiles de recursos limitados como netbooks,

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

dispositivos móviles y computadores embebidos. Con este ideal y esquema de desarrollo que posee, se presenta como una competencia a Xubuntu.



Figura 4.3 Fusión y resultado de lubuntu

Fuente: Gstatic (2013). Recuperado de https://encrypted-tbn1.gstatic.com/images?q=tbn:ANd9GcQa1zcSsuv7okQ_I7E1NNuq4G5xdfAXhcll44nMc26wKEVNa44U

4. DESARROLLO

1. Crear la SD arrancable

He aquí los pasos para crear una SD arrancable:

a) Descargar la imagen de Raspbian de la siguiente URL:

<https://www.raspberrypi.org/downloads/>

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

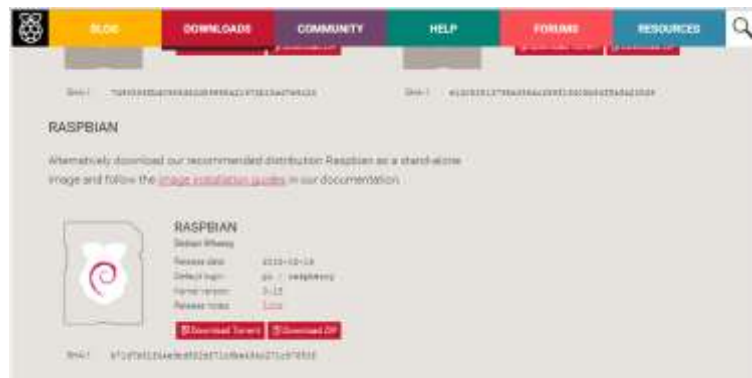


Figura 4.4 Sitio de descarga Raspberry

Fuente: Raspberry Pi Foundation (2015). Recuperado de <https://www.raspberrypi.org/downloads/>

2. La imagen está comprimida en un archivo ZIP.



Figura 4.5 Archivo Imagen

Fuente: Raspberry Pi Foundation (2015). Recuperado de <https://www.raspberrypi.org/downloads>

3. Descomprimir la imagen y obtener un archivo con extensión img.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

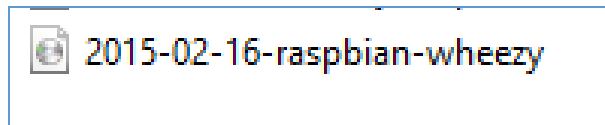


Figura 4.6 Archivo extensión.

- Introducir la tarjeta SD en el PC y crear la SD arrancable a partir del archivo de imagen, para ello utiliza la herramienta.

Win32DiskImager: <http://sourceforge.net/projects/win32diskimager>

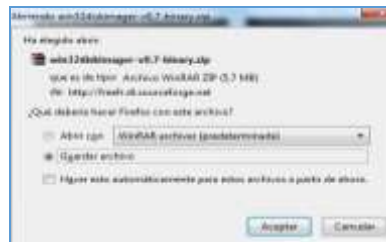


Figura 4.7 Descarga Win32DiskImager

- Descomprimir el programa Win32 Disk Imager que no necesita instalación.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

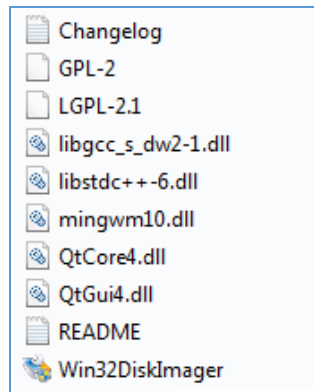


Figura 4.8 Descompresión Win32DiskImager

- Ejecutar (Win32DiskImager.exe) con propiedades de administrador si está usando Windows 7 o Windows 8.

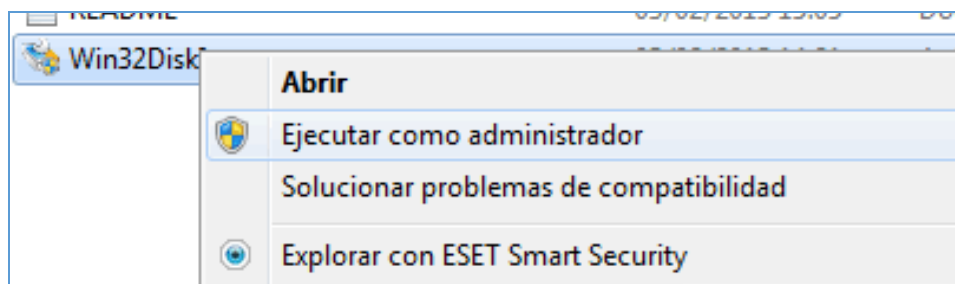


Figura 4.9 Abrir Win32DiskImager

Este es el entorno del programa Win32 Disk Imager (ver imagen 9):

En Win32 Disk Imager seleccionar la imagen del sistema operativo indicando la ruta donde se ha dejado al descomprimirlo mediante el botón con el icono de carpeta

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

azul. Después, asegurar que en Device está seleccionada la unidad en la que está insertada la tarjeta SD. Para comenzar el proceso de instalación en la tarjeta, pulsar el botón Write.

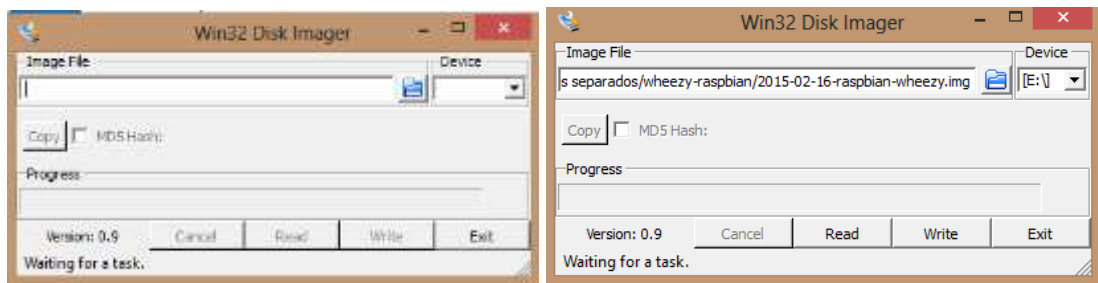


Figura 4.10 Entorno Win32DiskImager

- Se indicará que el contenido del dispositivo en cuestión será eliminado. Aceptar pulsando el botón Yes.

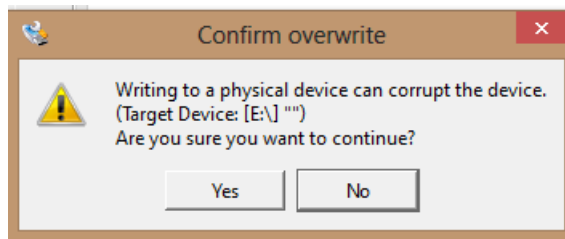


Figura 4.11 Ventana sobrescribir

- Ver la ventana de progreso.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

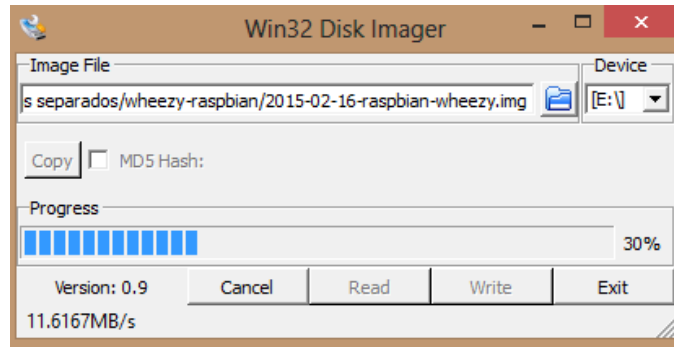


Figura 4.12 Progreso de avance.

9. Una vez terminado el progreso de instalación, se notifica con el mensaje de la imagen, y hay que aceptar.

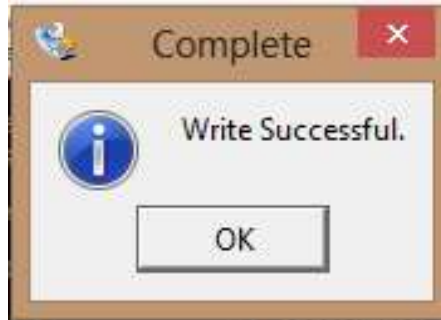


Figura 4.13 Indicación de terminación de Proceso

10. Ya se puede sacar la tarjeta SD de nuestro PC y ponerla en el zócalo de la Raspberry PI.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

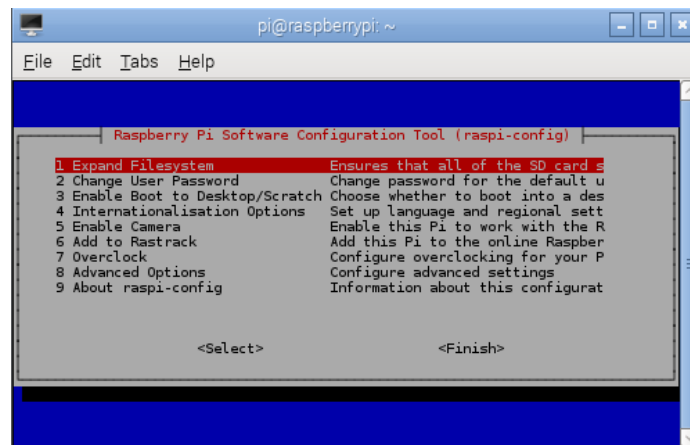
		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

11. Conectar el cable de Ethernet para dar acceso a Internet y alimentar la placa, conectar el teclado, el mouse y el monitor de video HDMI o el monitor VGA con su respectivo convertidor de formato,.
12. Después que Raspbian “Wheezy” ha sido instalado, se tiene que configurar el Sistema operativo para que funcione en español. Los siguientes pasos le muestra las diferentes opciones disponibles.
13. Al reiniciar el dispositivo el primer programa que se ejecuta se llama raspi-config, este programa solo se ejecuta en inglés. En caso de que ya haya instalado el sistema operativo y desee realizar alguna de estas modificaciones, lo puede hacer ejecutando el siguiente comando desde la terminal:

sudo raspi-config

14. Menú principal Raspberry PI

El primer menú que muestra contiene 9 diferentes opciones disponibles, a continuación se revisa cada una de las opciones disponibles:



Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:


		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Figura 4.14 Opciones Menú Raspberry

Opción 1

15. Expandir el sistema de archivos (Expand Filesystem)

Esta opción permite expandir el sistema operativo para que utilice todo el espacio disponible en la tarjeta. Cuando se instala Raspbian “Wheezy” la imagen copiada en la tarjeta solo ocupa 2 GB, por lo tanto es necesario ejecutar esta opción para que todo el espacio de la tarjeta SD sea utilizado.

Si el sistema operativo fue instalado utilizando NOOBS, no es necesario ejecutar esta opción. Ya el sistema operativo ha sido expandido.

Opción 2

16. Cambiar la contraseña del usuario Pi (Change User Password)

En el Raspberry Pi y en general en sistemas Linux existen diferentes tipos de usuario, los dos que vienen predeterminados por el sistema son los usuarios “root” y “pi”

El más importante que es el administrador del sistema que se llama “root”, este tiene acceso privilegiado a todos los archivos, configuraciones y carpetas del sistema. El otro tipo de usuario son los comunes como lo es “pi”, este viene predeterminado con la contraseña “raspberry” por lo tanto cualquier persona podría acceder su sistema.

Por eso, es recomendable cambiar la contraseña en esta opción. El sistema le solicitará que ingrese la nueva contraseña y que la repita nuevamente, al finalizar espere un mensaje como el siguiente:

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	



Figura 4.15 Cambio de contraseña

Opción 3.

17. Activar el escritorio al iniciar (Entable Boot to Desktop)

Esta opción permite que el Raspberry Pi después de iniciar el sistema, comience inmediatamente el escritorio modo gráfico o en línea de comando. En caso que inicie en modo de comando y después desee ingresar al modo gráfico solo ingrese el siguiente código:

- Startx

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

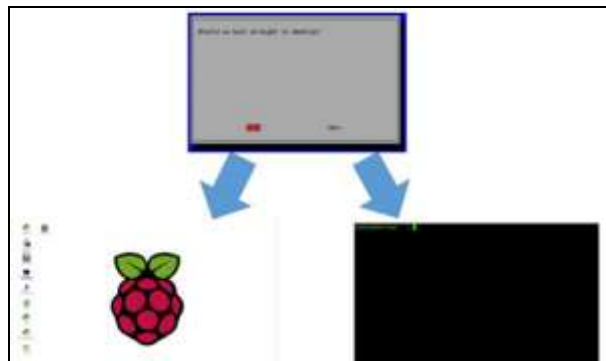


Figura 4.16 Cambio al modo gráfico

Opción 4.

18. Opciones de internacionalización (Internationalisation Options)

Esta opción permite modificar el lenguaje del sistema operativo, la zona horaria y la distribución de su teclado. Para este ejemplo se considera que se encuentra en Ecuador y tiene teclado en modo latinoamericano.

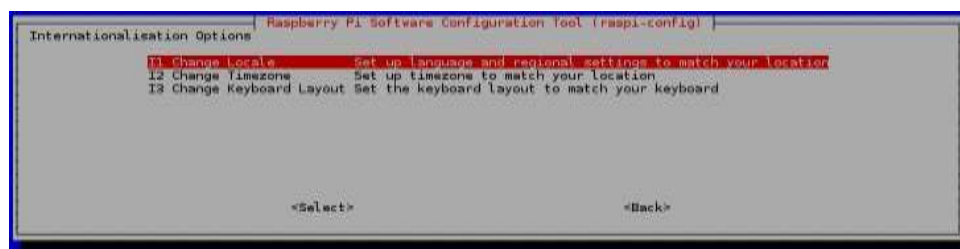


Figura 4.17 Cambio al modo gráfico

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

La opción I1 sirve para indicar donde se encuentra ubicado, esta opción configura el lenguaje del sistema operativo, los caracteres, la denominación de la moneda, etc. Busque y seleccione el modo de codificación:

- es_EC.UTF-8 UTF-8

Para confirmar oprima espacio y luego “enter”.

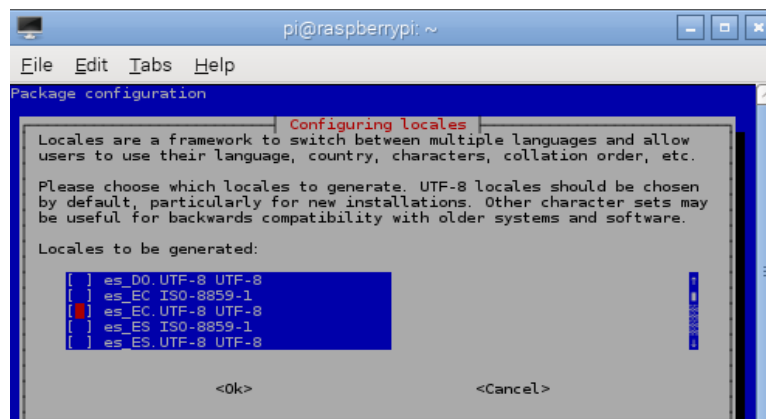


Figura 4.18 Selección opción I1

La opción I2 sirve para cambiar la zona horaria de su sistema la cual se ajusta de acuerdo a la ciudad donde vive o la más cercana. En las siguientes imágenes el sistema ha sido configurado como Ecuador->Guayaquil

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

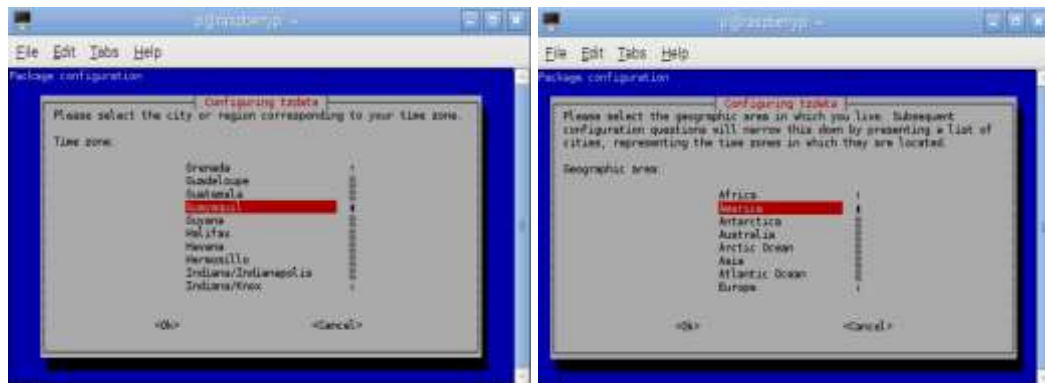


Figura 4.19 Selección opción I2

La opción A3 – le permitirá cambiar la configuración de su teclado, si la marca y tipo de su teclado no aparece, seleccione el predeterminado “PC genérico 105 teclas (intl)”.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

19. Activar la cámara (Enable camera)

Esta opción sirve para dar soporte a la cámara de Raspberry Pi, esta opción permite activar el puerto para que haya comunicación entre la CPU y el controlador de la cámara.

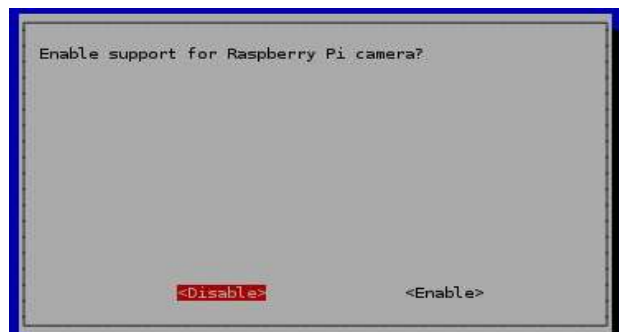


Figura 4.21 Habilitación de la cámara

Opción 6

20. Adicionar a Rastrack (Add to Rastrack)

Esta opción permite que su Raspberry Pi sea rastreado por el sitio web “Rastrack” (<http://rastrack.co.uk>), este sitio no pretende registrar o recolectar información alguna. Es una herramienta para tener la estadística de donde se encuentran los Raspberry Pi en el mundo. Es solo por diversión. Si desea que sea rastreado, solamente ingrese un apodo para identificar el Raspberry Pi y su dirección de correo.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	



Figura 4.22 Rastrack

Opción 7

21. Overclock

Esta opción permite aumentar la velocidad del procesador. Tenga en cuenta lo siguiente al modificar la velocidad. Primero, la vida del dispositivo se puede disminuir considerablemente. Segundo, el dispositivo generara más calor, por lo tanto es recomendable tener disipadores en el procesador, en circuito de Ethernet y en regulador de energía. Tercero, va a necesitar una fuente de poder de mayor capacidad para que pueda compensar la nueva velocidad seleccionada. Se recomienda dejar la predeterminada, pero si quiere tener más poder de cómputo siéntase libre de escoger la que más le convenga.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

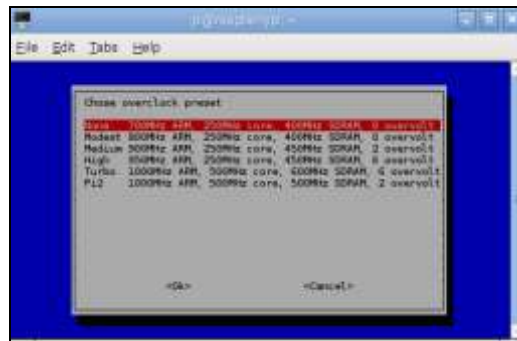


Figura 4.23 Selección Overclock

Opción 8.

22. Opciones avanzadas (Advanced Options)

Esta opción presenta otro submenú con las siguientes opciones.

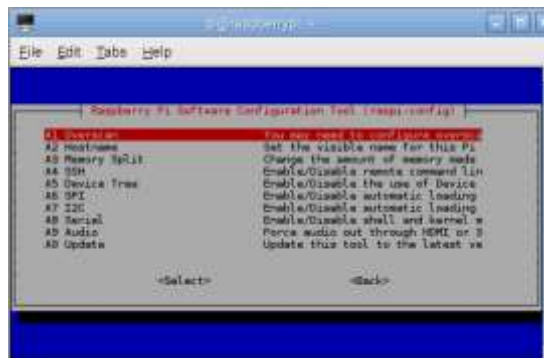


Figura 4.24 Menú Opciones Avanzadas

La opción A1 “overscan” sirve para borrar las líneas negras en algunos monitores o televisores.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

La opción A2 Hostname, sirve para identificar su Raspberry Pi en su red local, solamente utilice letras y números. Tenga en cuenta que el sistema diferencia mayúsculas y minúsculas, por lo tanto recuerde como escribe este nombre.

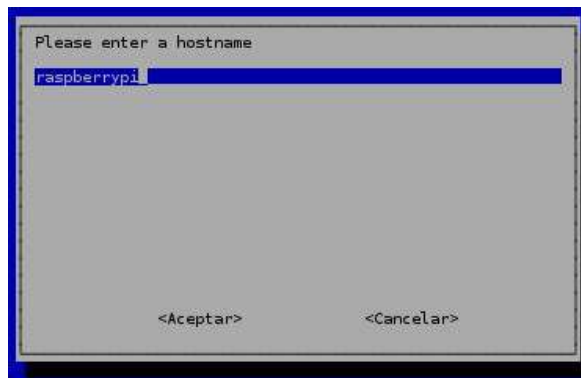


Figura 4.25 Opción A2 Hostname

La opción A3 – Distribución de la memoria (Memory Split) le permite seleccionar la cantidad de memoria compartida entre la CPU y la unidad de gráficos (GPU), el modelo B versión 2.0 cuenta con 512 MB en total. El predeterminado es 64 MB para la memoria de vídeo, si piensa ejecutar aplicaciones que requieren alto procesamiento gráfico como ver películas o ver imágenes aumente el valor. Tenga en cuenta que al aumentar la memoria de vídeo y disminuir la del procesador, este se vuelve un poco más ‘lento’ para ejecutar otras tareas, es el precio que toca pagar por ejecutar mejor los gráficos.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

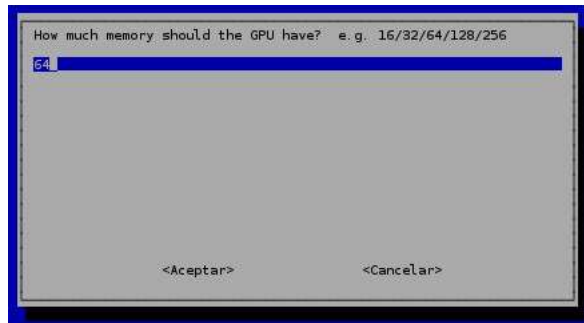


Figura 4.25 Opción A3 Distribución de Memoria

La opción A4 – Activar SSH (Enable SSH) se utiliza para acceder el Raspberry Pi remotamente desde un cliente SSH. SSH significa “Secure SHell” el cual es una forma segura de conectarse al Raspberry Pi a través de la red, es recomendable activar esta opción, ya que con esto no necesitará utilizar ni un monitor, ni teclado, ni mouse adicionales para poder controlar su dispositivo. En futuros tutoriales se enseña como utilizar esta opción.

La última opción A5 – Actualizar (update) se utiliza para que descargue una actualización del sistema, si ya se encuentra conectado a la red lo puede ejecutar inmediatamente. Si hay nuevas versiones de las librerías o programas se descargarán e instalarán las últimas versiones. O si prefiere lo puede hacer más adelante con el siguiente comando:

Sudo apt-get update

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Finalizado los pasos anteriores, correspondientes a la configuración inicial, es conveniente reiniciar la Raspberry.

Esta ejecutará el proceso de arranque y se detendrá en el modo consola o en el modo gráfico según lo que se le haya programado en la opción 3 del menú principal de “Raspi-config”.

Si se detiene en modo consola, pedirá que se ingrese el LOGIN y el Password correspondientes los que por defecto son los siguientes:

- login : pi
- Password: raspberry

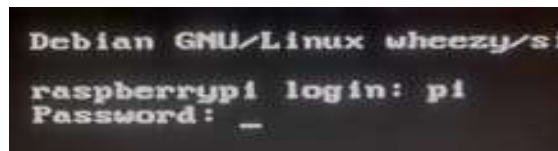


Figura 4.26 Ingreso de “Login-Password”

Luego ejecutar el siguiente comando para arrancar el modo gráfico:

- startx

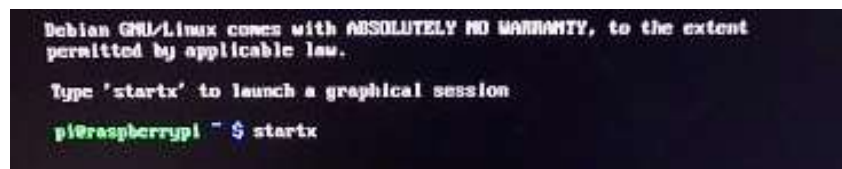


Figura 4.27 Ejecución modo Gráfico.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Como pasos finales actualizar los repositorios y el firmware de nuestra raspberry, ejecutando los siguientes comandos uno por uno en su orden respectivo:

- sudo apt-get update
- sudo apt-get upgrade
- sudo rpi-update

Esperar a que finalice la ejecución de cada uno, este procedimiento tarda algunos minutos para cada comando en la raspberry tipo B y puede ser realizado desde la línea inicial de comandos en el modo consola, o desde lxterminal en el modo gráfico.

Finalizado este procedimiento, reiniciar la raspberry con el siguiente comando:

- sudo reboot

Arrancando la raspberry, se pedirá nuevamente “login y password”, que por defecto ya se lo conoce (login: pi, password: raspberry), y queda listo para trabajar con la misma.

- ODROID-U3:

1. Crear la SD arrancable

Los pasos a seguir para la creación de la SD arrancable son similares a los empleados para la raspberry-pi.

He aquí los pasos para crear una SD:

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Descargar la imagen de lubuntu de la siguiente URL:

[img.http://odroid.com/dokuwiki/doku.php?id=en:u3_release_linux_ubuntu](http://odroid.com/dokuwiki/doku.php?id=en:u3_release_linux_ubuntu)

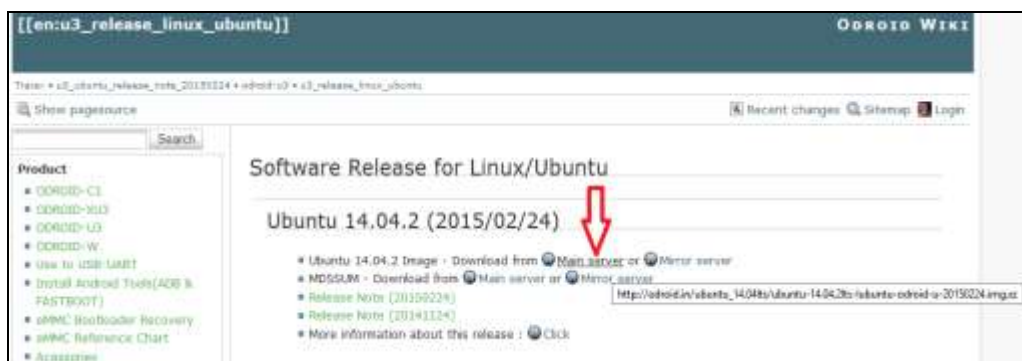


Figura 4.28 Sitio descarga Lubuntu.

- La imagen está comprimida en un archivo ZIP. Descomprimir la imagen. Obtener un archivo con extensión img.

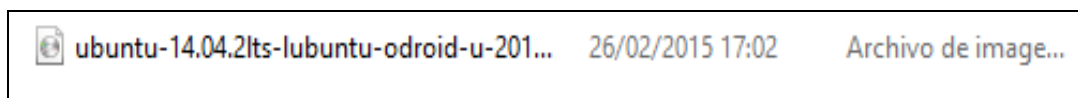


Figura 4.29 Archivo extensión.

- Introducir la tarjeta SD en el PC y crear la SD arrancable a partir del archivo de imagen, utilizando la herramienta.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Win32DiskImager: <http://sourceforge.net/projects/win32diskimager>. Este paso es similar al empleado en la raspberry-pi.

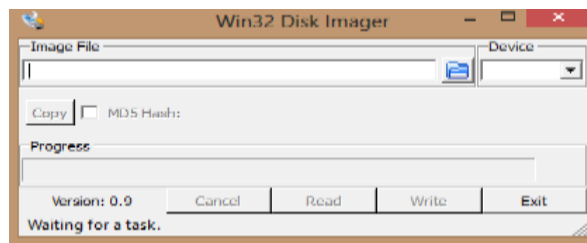


Figura 4.30 Entorno Win32DiskImager

- En Win32 Disk Imager seleccionar la imagen del sistema operativo indicando la ruta donde se deja al descomprimirlo mediante el botón con el icono de carpeta azul. Después, se asegura que en Device está seleccionada la unidad en la que está insertada la tarjeta SD. Para comenzar el proceso de instalación en la tarjeta, pulsar el botón Write. Se le indica que el contenido del dispositivo en cuestión va a ser eliminado. Aceptar pulsando el botón Yes.

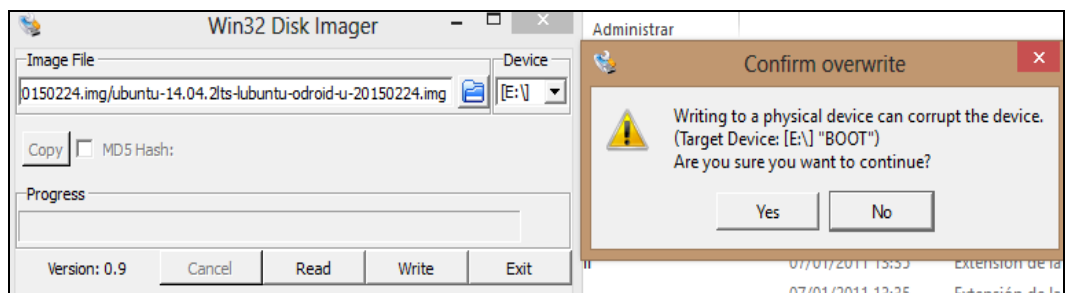


Figura 4.31 Confirmación de ruta y sobre escritura

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

5. Ver la ventana de progreso.

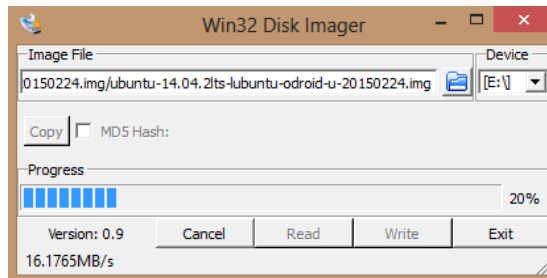


Figura 4.32 Progreso de avance.

6. Una vez terminado el progreso de instalación, se le notifica con el mensaje de la imagen. Dar click en OK para aceptar.

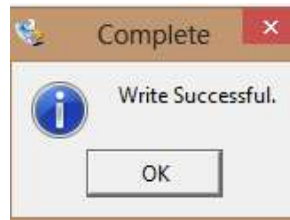


Figura 4.33 Indicación de terminación de Proceso

7. Ya se puede sacar la tarjeta SD de nuestro PC y ponerla en el zócalo de la ODROID-U3
8. Conectar el cable de Ethernet para dar acceso a Internet y alimentar la placa, conectar el teclado, el mouse y el monitor de video HDMI o el monitor VGA con su respectivo convertidor de formato.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

9. Después que Ubuntu ha sido instalado, configurar el Sistema operativo para que funcione en español. Los siguientes pasos le muestra las diferentes opciones disponibles.

10. Configuración de Ubuntu.

A diferencia de “Raspbian” en la Raspberry-pi, Ubuntu arranca directamente en un entorno grafico conocido como LXDE.



Figura 4.34 Entorno gráfico LXDE

En la OdROID-U3 al arrancarla inicialmente hacer “click” en el icono Odroid utility y seguir los siguientes pasos para su configuración inicial:

Paso 1.

11. Hacer click en el icono ODROID Utility, se abre la ventana conocida como Terminal y le solicita la clave que por defecto es: odroid.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	



Figura 4.35 Ingreso a Odrid Utility

12. En la pantalla aparecerá el menú de configuración, hacer “click” en la opción 1, “HDMI configuration” y seleccionar la opción número 4 que permite trabajar con una resolución estándar para nuestro convertidor HDMI to VGA.

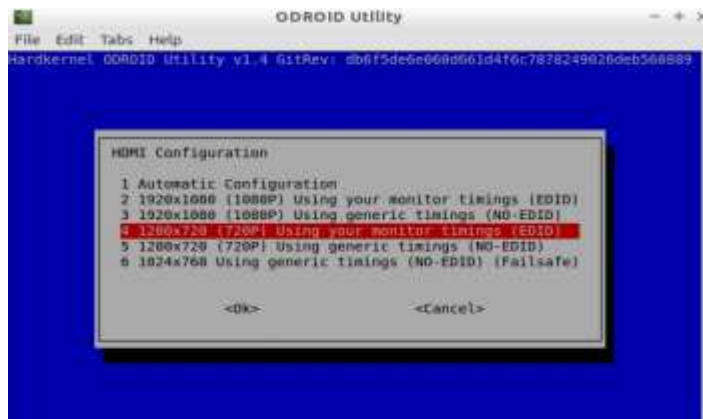


Figura 4.36 Selección salida HDMI.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

13. Hacer “click” a “Resize your root partition” opción 4, para expandir el sistema operativo a toda la capacidad de nuestra SD.

14. Luego actualizar el Kernel/Firmware por medio de la selección de la opción 2.

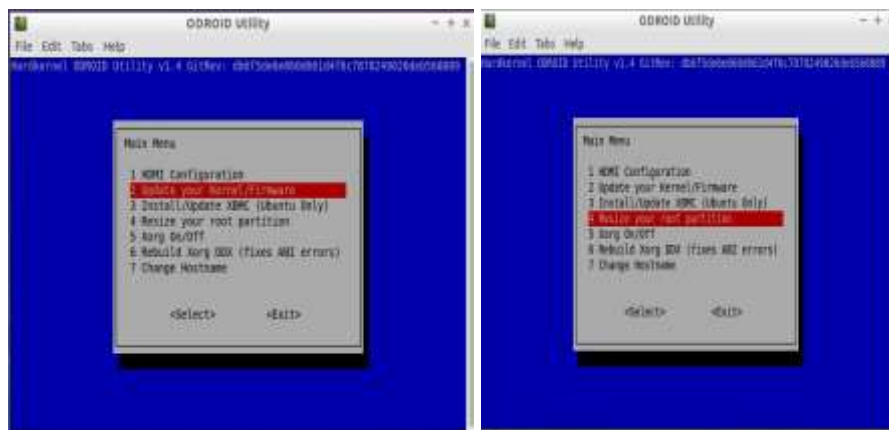


Figura 4.37 Actualización Kernel

15. Las otras opciones del menú de configuración son similares a sus homologas que se encuentra en la Raspberry –Pi, vienen configuradas por defecto y no es necesario modificarlas.

16. Reiniciar el sistema para que se apliquen los cambios de configuración.

Paso 2.

17. Para configurar el idioma abrir Lxterminal y ejecutar el siguiente comando:

- `sudo apt-get install console-data unicode-data language-pack-es-base`

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

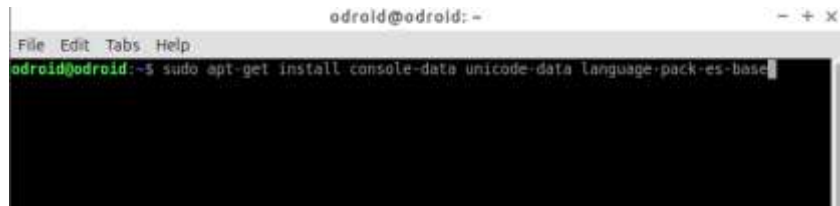


Figura 4.38 Configuración del idioma

18. En el proceso se le pregunta el tipo de teclado que va a utilizar:



Figura 4.39 Selección tipo de Teclado

19. Escoger select keymap from full list y luego:

pc/qwerty/Spanish/Standard/Standard

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

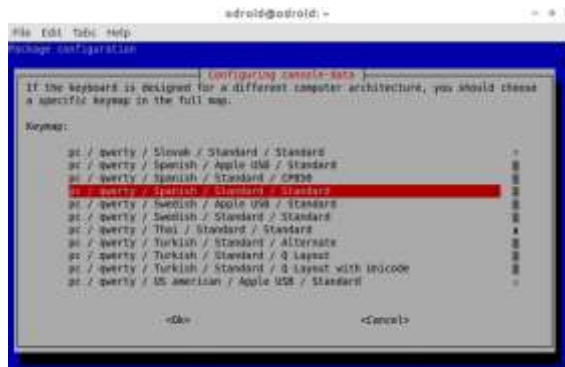


Figura 4.40 Selección tipo de Teclado

20. Después de esto modificar el archivo local , ejecutando la siguiente línea de instrucción en Lxterminal donde se indica la ruta del mismo para emplear el editor de archivos nano:

- `sudo nano /etc/default/locale`

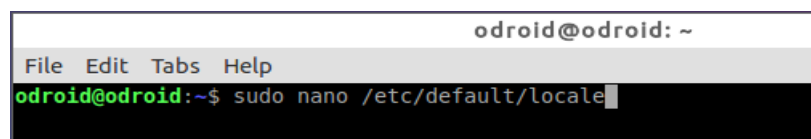


Figura 4.41 Ruta para modificación de archivo

Luego modificar el archivo `/etc/default/locale` para dejarlo así:

`LANG="es_EC.UTF-8"`

`LANGUAGE="es"`

`LC_NUMERIC="es_EC.UTF-8"`

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

LC_TIME="es_EC.UTF-8"

LC_MONETARY="es_EC.UTF-8"

LC_PAPER="es_EC.UTF-8"

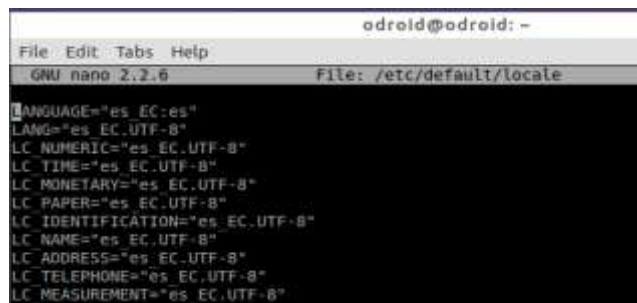
LC_IDENTIFICATION="es_EC.UTF-8"

LC_NAME="es_EC.UTF-8"

LC_ADDRESS="es_EC.UTF-8"

LC_TELEPHONE="es_EC.UTF-8"

LC_MEASUREMENT="es_EC.UTF-8"



```

odroid@odroid: ~
File Edit Tabs Help
GNU nano 2.2.6 File: /etc/default/locale
LANGUAGE="es_EC:es"
LANG="es_EC.UTF-8"
LC_NUMERIC="es_EC.UTF-8"
LC_TIME="es_EC.UTF-8"
LC_MONETARY="es_EC.UTF-8"
LC_PAPER="es_EC.UTF-8"
LC_IDENTIFICATION="es_EC.UTF-8"
LC_NAME="es_EC.UTF-8"
LC_ADDRESS="es_EC.UTF-8"
LC_TELEPHONE="es_EC.UTF-8"
LC_MEASUREMENT="es_EC.UTF-8"

```

Figura 4.42 Modificación Archivo Locale

21. Presionar control+x y seguidamente presionar la letra Y (yes) y luego enter para guardar los cambios.

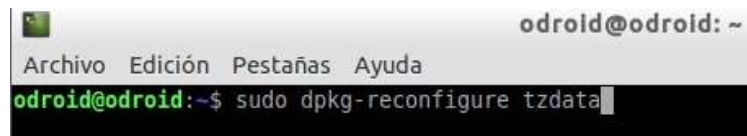
Paso 3.

22. Ahora configurar la fecha y hora, esto se hace fácilmente ejecutando en Lxterminal el siguiente comando:

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

- sudo dpkg-reconfigure tzdata



```
odroid@odroid: ~
Archivo Edición Pestañas Ayuda
odroid@odroid:~$ sudo dpkg-reconfigure tzdata
```

Figura 4.43 Comando cambio de fecha y hora

23. Ahora configurar el lugar geográfico o división política del lugar donde se encuentra. Escoger América, Guayaquil.

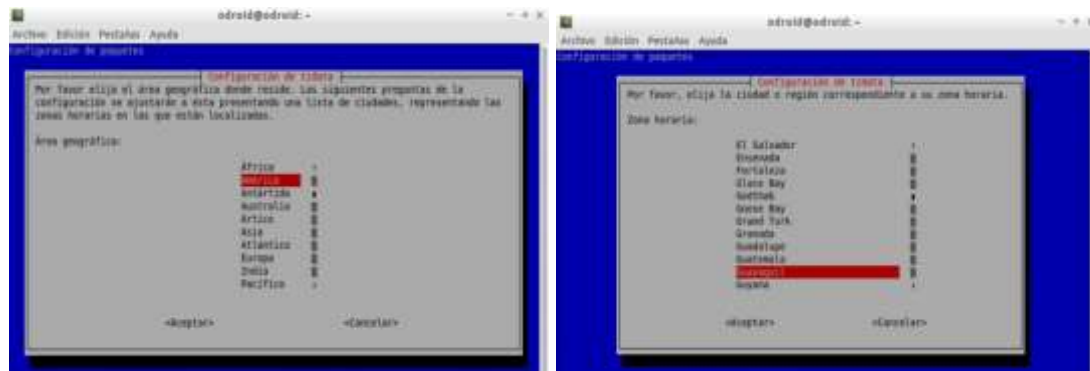


Figura 4.44 Selección de ubicación geográfica

24. Finalmente reiniciar el sistema y este quedará listo para funcionar con la configuración en español.

25. Instalación de programas en Raspberry-Pi (Raspbian) y ODROID-U3(Lubuntu)

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Es recomendable actualizar los repositorios y el firmware de los sistemas embebidos antes de realizar la instalación de un nuevo programa y también de manera periódica.

Para esto abrir una ventana terminal y ejecutar los siguientes comandos en orden respectivo en la Raspberry-pi:

- sudo apt-get update
- sudo apt-get upgrade
- sudo rpi-update

Y en ODROID-U3:

- sudo apt-get update
- sudo apt-get dist-upgrade
- sudo apt-get autoremove

La instalación de los diversos programas que se pueden correr en los sistemas operativos que utilizan Raspberry-pi y ODROID-U3 es sencilla, basta con abrir una ventana Terminal y ejecutar el siguiente comando:

- sudo apt-get install (nombre del programa)

Por ejemplo para instalar motion (programa empleado en video vigilancia) bastara con ejecutar en la terminal:

- sudo apt-get install motion

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 1
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	


A continuación se detalla una lista de los programas y utilidades que se emplearan en la Raspberry-Pi y en ODROID-U3:

- motion
- mutt
- xrdp
- festival
- scilab
- octave
- qt-creator
- qt4-dev-tools
- arduino
- kicad

5. CONCLUSIONES.

Al terminar esta práctica el alumno de Ingeniería Electrónica debe establecer establecer sus conclusiones.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 2
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

PRÁCTICA No 2

NÚMERO DE ESTUDIANTES: 20

ING. LUIS CÓRDOVA

TIEMPO ESTIMADO 2 HORAS


TEMA:

**CONFIGURACIÓN DE LA RED INALÁMBRICA Y SISTEMAS DE VÍDEO
VIGILANCIA.**



Figura 4.45 Logo software Motion.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 2
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

1. OBJETIVO GENERAL.

- Enseñar al estudiante la configuración a realizar para lograr la conexión de los sistemas embebidos a redes inalámbricas disponibles.

2. OBJETIVOS ESPECÍFICOS.

- Como modificar los archivos de configuración para la Raspberry PI y la ODROID-U3.
- Familiarizar al estudiante con el uso del programa MOTION.
- Como utilizar los sistemas Embebidos para aplicaciones de vídeo vigilancia.


3. MARCO TEÓRICO.

¿Qué es MOTION?

Motion, *un software detector de movimiento*, es un programa libre, de código abierto utilizado en aplicaciones de CCTV desarrollado para Linux.

Se puede monitorear la señal de vídeo de una o más cámaras y es capaz de detectar si una parte significativa de la imagen ha cambiado, también se puede configurar para que se tomen acciones de interacción con otros programas (comandos configurables) cuando detecta que el movimiento se está produciendo (también puede hacer unos vídeos de lapso de tiempo, etc.).

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 2
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

El programa está escrito en C y está hecho para Linux (explotando la interfaz video4linux). Motion es una herramienta basada en la línea de comandos cuya salida puede ser jpeg , bmp o mpeg en secuencias de vídeo.

Se opera principalmente a través de los archivos de configuración, aunque los flujos de vídeo se pueden ver desde un navegador web. También puede el usuario llamar a "disparadores" configurables cuando ocurren ciertos eventos.

4. DESARROLLO.

- Instalación del modem TP-LINK TL-WN725N en la Raspberry-Pi

Para utilizar el adaptador USB WiFi TP-Link TL-WN725N es necesario instalar el driver. Para eso, compilar. Todos los comandos se ejecutaran en Lxterminal.

Estos son los pasos que se deben seguir para hacerlo.

- Averiguar el kernel que tiene en la Raspberry Pi :

```
sudo uname -a
```


El resultado será algo como esto:

```
Linux raspberry Pi 3.12.35+ #730 PREEMPT Mon Sep 8 15:28:00 BST 2014 armv6l
GNU/Linux
```

De esta información tener en cuenta lo siguiente 3.12.35+ #730 pues con esta información seleccionar de la lista la URL de descarga del controlador

- Ahora ir al siguiente link:

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 2
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

<http://www.raspberrypi.org/forums/viewtopic.php?f=28&t=62371>

Para poder conocer cuál es fragmento de URL de descarga, de acuerdo con la información anterior (3.12.35+ #730)

```

CODE: SELECT ALL
3.12.29+ #714, #715 - 8188eu-20140908.tar.gz
3.12.30+ #717      - 8188eu-20141017.tar.gz
3.12.31+ #718, #720 - 8188eu-20141026.tar.gz
3.12.32+ #721      - 8188eu-20141107.tar.gz
3.12.33+ #722, #724 - 8188eu-20141107.tar.gz
3.12.34+ #725, #727 - 8188eu-20141107.tar.gz
3.12.35+ #730, #733 - 8188eu-20141107.tar.gz

```


Figura 4.46 Fragmento URL

Con la información que esta subrayada, completar la URL quedando así <https://dl.dropboxusercontent.com/u/80256631/8188eu-20141107.tar.gz>, de esta URL la información que siempre varia es la que se encuentra después de ultimo slash, esta URL la necesita para el completar el primer comando del paso siguiente.

Ahora descargar e instalar el controlador:

- `wget https://dl.dropboxusercontent.com/u/80256631/8188eu-20141107.tar.gz`
- `tar -zxvf 8188eu-20141107.tar.gz`

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 2
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

- `sudo install -p -m 644 8188eu.ko
/lib/modules/3.12.35+/kernel/drivers/net/wireless`
- `sudo insmod /lib/modules/3.12.35+/kernel/drivers/net/wireless/8188eu.ko`
- `sudo depmod -a`
- `sudo reboot`

En el caso del tercer y cuarto comandos, es necesario cambiar 3.12.35+ por el valor que le aparece cuando se ejecuta el comando del paso 1

Al haber ejecutado los comandos anteriores y después del reinicio que se generó en el último comando, proceder a hacer la conexión wifi, pues ya debe haber reconocido el modem TP-LINK WN-25N (se encontrará titilando el led azul del modem) y se observa la lista de redes inalámbricas disponibles.


Para la ODROID-U3 no es necesario instalar los drivers correspondientes al modem TP-LINK WN-722N, puesto que el sistema operativo Lubuntu lo reconoce inmediatamente.

Usar una IP estática

Es muy recomendable darle una IP estática a nuestros sistemas Odroid-U3 y Raspberry-Pi para poder localizarlos siempre. Como en cualquier otra distribución de Linux configurar el correspondiente archivo *interfaces* ejecutando la siguiente línea de comando en Lxterminal:

```
sudo nano /etc/network/interfaces
```

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 2
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Localizar la línea que pone: *iface eth0 inet dhcp* y se cambia por las siguientes:

```
iface eth0 inet static
address 192.168.1.13
netmask 255.255.255.0
gateway 192.168.1.1
```

Donde en address se coloca la IP que se desea que tenga siempre y en gateway la dirección del router, aunque normalmente suele ser la indicada. Además en auto se quita lo que haya y se pone auto eth0.

El resultado tiene que ser como este.




```
GNU nano 2.2.6 File: /etc/network/interfaces
auto eth0

iface lo inet loopback
iface eth0 inet static
address 192.168.1.13
netmask 255.255.255.0
gateway 192.168.1.1

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Figura 4.47 Uso de Ip estática.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 2
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Interfaces (5) file used by ifup(8) and ifdown(8)

auto eht0

iface lo inet loopback

iface eth0 inet static

address 192.168.1.10

network 255.255.255.0

gateway 192.168.1.1

Configurar usuarios en ODROID-U3

Odroid viene por defecto con usuario “odroid”. Se puede crear otro usuario personal y eliminar el usuario odroid o cambiar su contraseña.

- `sudo adduser (nombre_usuario)`


El nuevo usuario debe ser agregado a la lista de usuarios sudo para que pueda ejecutar comandos como root. Esto se realizará así:

- `sudo usermod -a -G sudo (nombre_usuario)`

Una vez agregado el nuevo usuario, el usuario anterior: ODROID puede ser eliminado:

- `sudo deluser --remove-home odroid`

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 2
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Si el home de odroid no se borra se lo puede eliminar así:

- `sudo rm -rf /home/odroid`

Reiniciar y listo.

SISTEMAS DE VÍDEO VIGILANCIA

A continuación indico los pasos que se deben seguir para instalar Motion:

Instalar V4L (Video para Linux) si no se tiene instalado:

- `sudo apt-get install v4l-utils`

Instalar motion:

- `sudo apt-get install motion`

Listar los dispositivos USB que se tienen conectados y comprobar que nuestra cámara se encuentra entre ellos:

- `lsusb`

Visualizar las características de nuestra cámara:

- `sudo v4l2-compliance -d /dev/video0`


Confirmar que se tiene un dispositivo de video conectado:

- `ls -al /dev/video*`

Configurar motion:

- `sudo nano /etc/motion/motion.conf`

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 2
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

En la siguiente URL:

<http://www.lavrsen.dk/foswiki/bin/view/Motion/ConfigFileOptions>

Encontrar información detallada acerca de cada uno de los parámetros que pueden ser modificados en el archivo de configuración. Este programa es un proyecto de libre desarrollo que permite obtener sistemas de video vigilancia y seguridad.

A continuación se muestran los parámetros que se han modificado:

```

041 bash 2.2.6: P04090: /etc/motion/motion.conf
# Rename this distribution example file to motion.conf
#
# This config file was generated by motion 3.2.12

#####
# libeexon
#####

# Start in daemon (background) mode and release terminal (default: off)
daemon on

# File to store the process ID, also called pid file. (default: not defined)
process_id_file /var/run/motion/motion.pid


#####
# Basic Setup Mode
#####

Ver ayuda  Guardar  Leer fich  Pag Ant  ContarTex  Fin actual
Salir  Justificar  Borrar  Pag Sig  PagarTex  Ortografía

```

Figura 4.48 daemon on (por defecto está off)

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

REVISIÓN 1/1	
	PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 2
LABORATORIO	ELECTRÓNICA-EMBEBIDOS
CARRERA	INGENIERÍA ELECTRÓNICA
SEDE	GUAYAQUIL

```

GNU nano 2.2.6: /etc/motion/motion.conf
# Output frames at 1 fps when no motion is detected and increase to the
# rate given by webcam_maxrate when motion is detected (default: off)
webcam_motion off

# Maximum framerate for webcam streams (default: 1)
webcam_maxrate 1

# Restrict webcam connections to localhost only (default: on)
webcam_localhost off

# Limits the number of images per connection (default: 0 = unlimited)
# Number can be defined by multiplying actual webcam rate by desired number of 5
# Actual webcam rate is the smallest of the numbers framerate and webcam_maxrate
webcam_limit 0

#####
# HTTP Based Control

```

Figura 4.49 webcam_localhost off (por defecto está on)

```

GNU nano 2.2.6: /etc/motion/motion.conf
# Wx = event, Wl = frame number, Wt = thread (camera) number,
# Wc = changed pixels, Wn = noise level, Ws = new line,
# Ww and Wh = width and height of motion area,
# Wx and Wl = X and Y coordinates of motion center
# Wc = value defined by text_event - do not use with text_event!
# You can put quotation marks around the text to allow
# leading spaces.
#####
# Locate and draw a box around the moving object.
# Valid values: on, off and preview (default: off)
# Set to "preview" will only draw a box in preview_shot pictures.
locate on

# Draws the timestamp using some options as C function strftime(3)
# Default: %Y-%m-%d%T = date in ISO format and time in 24-hour clock
# Text is placed in lower right corner
text_right %Y-%m-%d%T-%a

```

Figura 4.50 locate on (por defecto está off)

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 2
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	



Figura 4.51 width 640 y height 480

También se puede cambiar el puerto por defecto *8081*. El número de frames por segundos capturados *framerate*, por defecto 100, la calidad de las capturas, etc.

Después de realizar los cambios más convenientes, guardar **Ctrl+O** y salir **Ctrl+X**.

Dirigir a otro fichero de configuración de motion:

- `sudo nano /etc/default/motion`



Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:


		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 2
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Figura 4.52 Cambio de Puerto.

Cambiar de NO a YES, guardar los cambios y cerrar.

Ahora reiniciar el servicio:

- `sudo service motion restart`

Dirigir al navegador favorito:

- `http://ip-raspberry:puerto`

Por ejemplo, `http://192.168.1.19:8081` (si no se ha cambiado el puerto por defecto).

Y ya se tiene acceso al video en directo. La calidad de la imagen dependerá de la cámara que se está utilizando.




Figura 4.53 Captura de imagen.

5. CONCLUSIONES.

Al terminar esta práctica el alumno de Ingeniería Electrónica debe establecer sus conclusiones.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

PRÁCTICA No 3
NÚMERO DE ESTUDIANTES 20
ING. LUIS CÓRDOVA
TIEMPO ESTIMADO 2 HORAS

TEMA:

**CONFIGURACIÓN Y USO DE LOS PINES DIGITALES DEL PUERTO
GPIO.**

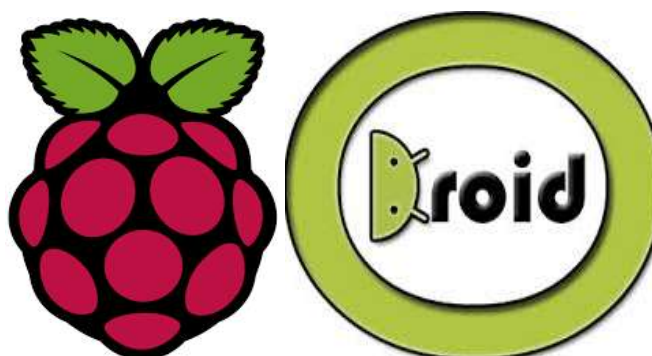



Figura 4.54 Logo de equipos empleados.

1. OBJETIVO GENERAL.

Conocer el funcionamiento y las aplicaciones de los pines GPIO en los sistemas embebidos.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

2. OBJETIVOS ESPECÍFICOS.

3. MARCO TEÓRICO.

Descripción básica del puerto GPIO

La tarjeta Raspberry Pi puede comunicarse con dispositivos externos mediante el conector GPIO incorporado. En dicho conector se integran patillas de alimentación (+5 y +3.3 V), masa, y entradas/salidas capaces de implementar diferentes protocolos.

Como quiera que haya 2 versiones de Raspberry a nivel hardware (Rev. 1 y 2) las asignaciones de puertos también varían. Sin entrar de momento en detalles de protocolos, las dos posibles versiones de nuestra Raspberry hacen que se encuentre a su vez con dos posibles escenarios hardware. Se observa aquí un esquema comparativo de la numeración de patillas.

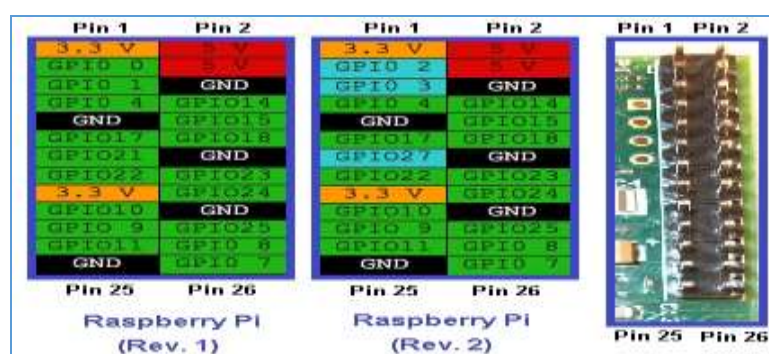



Figura 4.55 Pin out

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Es importante tener en cuenta que – a nivel software – se sabe qué patilla se quiere comunicar, y por tanto, en esta y en posteriores entradas se opta por utilizar patillas comunes a ambas releases.

Además de las patillas correspondientes a +5, +3.3 y masa, se tiene varios pines de uso genérico donde se puede conectar dispositivos hardware. En este artículo, de momento, se explica la forma de utilizar una patilla como entrada (INPUT) y otra como salida (OUTPUT).

Es muy importante comentar que cualquier manipulación errónea, conexionado equivocado o descarga estática sobre las patillas GPIO puede dañarlas de forma permanente. Por esta razón se aconseja seguir fielmente las recomendaciones dadas al respecto.

Cómo “ve” Linux el puerto GPIO de la Raspberry Pi.

Linux, como ya muchos conocen, referencia casi todo lo que maneja en forma de “fichero”, y las patillas del puerto GPIO no iban a ser una excepción. No es preciso que sean “expertos” en Linux y conocer exactamente su gestión interna, pero lo que si conviene saber es que el “núcleo” de Linux ya sabe de la existencia de dicho puerto, pero no así el resto del sistema Linux, razón por la que se tiene que “informar” a nuestra tarjeta (vía software) de la presencia del puerto GPIO para que la Raspi “vea” el mismo.

Para realizar esta operación se tiene que “exportar” la información I/O desde el núcleo Linux antes de poder acceder al puerto tal y como si de un fichero más se

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

tratase. Para ello efectuar los comandos destinados a “crear” el acceso al hardware vía ficheros, y una vez efectuados se tiene el acceso en /sys/class/gpio.

- Nota: la operativa de exportación ha de hacerse como root (su)

Si da un vistazo a dicho directorio antes de “mapear” puerto alguno se observa que el contenido es:

```

pi@raspberrypi ~ $ cd /sys/class/gpio/
pi@raspberrypi /sys/class/gpio $ ls -la
total 0
drwxr-xr-x  2 root root   0 abr 13 02:39 .
drwxr-xr-x 37 root root   0 abr 13 02:37 ..
--w-----  1 root root 4096 abr 13 03:31 export
lrwxrwxrwx  1 root root   0 abr 13 03:31 gpiochip0
--w-----  1 root root 4096 abr 13 03:31 unexport
pi@raspberrypi /sys/class/gpio $ █

```

Figura 4.56 Contenido directorio

Salir de este directorio

- cd

Ahora se adquiere privilegios de root para efectuar los comandos mediante

- sudo -i

y efectuar en primer lugar el comando para exportar la patilla gpio14 como salida mediante

```

root@raspberrypi:~# echo "14" > /sys/class/gpio/export
root@raspberrypi:~# chmod 777 -R /sys/class/gpio/gpio14
root@raspberrypi:~# echo "out" > /sys/class/gpio/gpio14/direction
root@raspberrypi:~# █

```

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:


REVISIÓN 1/1	
	PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS
CARRERA	INGENIERÍA ELECTRÓNICA
SEDE	GUAYAQUIL

Figura 4.57 Corrida de comandos

- echo "14" > /sys/class/gpio/export

le da permisos totales

- chmod 777 -R /sys/class/gpio/gpio14

Fijar la dirección en la patilla como "salida"

- echo "out" > /sys/class/gpio/gpio14/direction

Si ahora se observa de nuevo el contenido de /sys/class/gpio/ ae tiene:

```

root@raspberrypi:~# ls -la /sys/class/gpio/
total 0
drwxr-xr-x  2 root root  0 abr 13 02:39 .
drwxr-xr-x 37 root root  0 abr 13 02:37 ..
-w-----  1 root root 4096 abr 13 03:40 export
lrwxrwxrwx  1 root root  0 abr 13 03:41 gpio14 -> ../../devices/virtual/gpio/gpio14
lrwxrwxrwx  1 root root  0 abr 13 03:31 gpiochip0 -> ../../devices/virtual/gpio/gpiochip0
-w-----  1 root root 4096 abr 13 03:31 unexport
root@raspberrypi:~#

```


Figura 4.58 Revisión de contenido del directorio

por lo que ya se comprueba que gpio14 está "exportado"

Debe repetir un proceso homólogo con la patilla que se usa como entrada (gpio8), así:

- echo "8" > /sys/class/gpio/export
- chmod 777 -R /sys/class/gpio/gpio8
- echo "in" > /sys/class/gpio/gpio8/direction

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Y puede comprobar luego el efecto en el directorio /sys/class/gpio/

```

root@raspberrypi:~# echo "8" > /sys/class/gpio/export
root@raspberrypi:~# chmod 777 -R /sys/class/gpio/gpio8
root@raspberrypi:~# echo "in" > /sys/class/gpio/gpio8/direction
root@raspberrypi:~# ls -la /sys/class/gpio/
total 0
drwxr-xr-x  2 root root    0 abr 13 02:39 .
drwxr-xr-x 37 root root    0 abr 13 02:37 ..
---w-----  1 root root 4096 abr 13 03:49 export
lrwxrwxrwx  1 root root    0 abr 13 03:41 gpio14 -> ../../devices/virtual/gpio/gpio14
lrwxrwxrwx  1 root root    0 abr 13 03:50 gpio8 -> ../../devices/virtual/gpio/gpio8
lrwxrwxrwx  1 root root    0 abr 13 03:31 gpiochip0 -> ../../devices/virtual/gpio/gpiochip0
---w-----  1 root root 4096 abr 13 03:31 unexport
root@raspberrypi:~# █

```

Figura 4.59 Revisión de contenido del directorio

4. DESARROLLO.

Primer programa (sencillo) de prueba: lectura y escritura vía GPIO.

En la Raspberry Pi efectuar la exportación de puertos (gpio14 como salida y gpio8 como entrada) tal y como se ha comentado en el punto anterior.

Ahora ya está listo para leer y escribir en el puerto GPIO de nuestra Raspi.

Si ha realizado la “exportación” (y antes de enviar comando alguno desde la Raspi) deberá estar activo solo el LED amarillo (POWER). En este momento el relé mantiene cerrados los contactos “Co” y “NC”.

Para cambiar el estado de la salida – y por tanto el del relé, el LED verde indicador y cualquier dispositivo conectado a través del mismo – ejecutar:

echo “1” > /sys/class/gpio/gpio14/value

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

activando la salida y haciendo conmutar al relé. En este instante los contactos “Co” y “NO” deben haberse cerrado y el LED verde deberá activarse para desactivar la salida ejecutar.

echo “0” > /sys/class/gpio/gpio14/value

Ahora los contactos “Co” y “NC” deben haberse cerrado y el LED verde deberá apagarse.

```

root@raspberrypi:~# echo "1" > /sys/class/gpio/gpio14/value
root@raspberrypi:~# echo "0" > /sys/class/gpio/gpio14/value
root@raspberrypi:~# █

```

Figura 4.60 Activación desactivación de la salida.

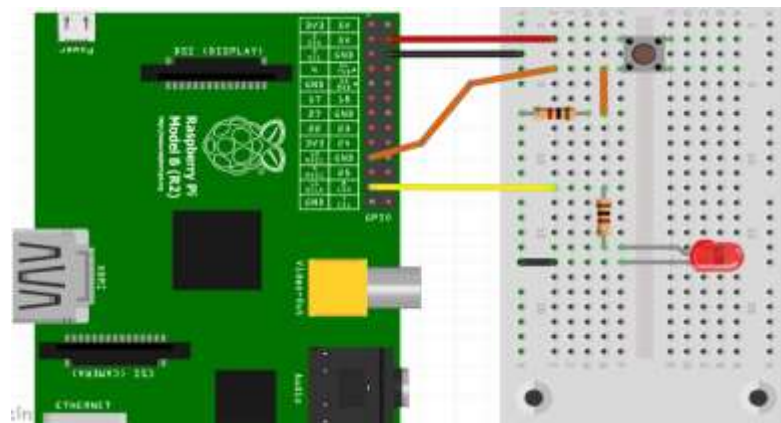



Figura 4.61 Armado de circuito de Prueba en el PROTOBOARD

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Se controla mediante los contactos del relé un dispositivo externo desde nuestra Raspberry. ¿Cuál? ... eso ya dependerá de nuestra imaginación.

Controlando la entrada a la Raspberry Pi –

Para “leer” esta entrada (se supone ya mapeado / exportado el puerto gpio8) observar el valor (0/1) que hay en el fichero exportado anteriormente, por lo que se ejecuta el comando:

- `more /sys/class/gpio/gpio8/value`

Se observa que el valor obtenido depende de que se pulse (1) o no (0) el micro-interruptor de “TEST”. De esta forma se puede crear un software que actúe dependiendo del estado de nuestra entrada...

```

root@raspberrypi:~# more /sys/class/gpio/gpio8/value
0
root@raspberrypi:~# more /sys/class/gpio/gpio8/value
1
root@raspberrypi:~# █


```

Figura 4.62 Cambio de estado en el programa.

ODROID U3.

Puede que quieras conectar algunos servomotores, LEDs y conmutadores a ODROID-U3. Pero sólo hay un conector de 8 pines, así que ¿cómo conseguir más? Las corrientes eléctricas son de 5V, 1.8V y tierra, dejando sólo 5 puertos para usarse. Aún más confuso es utilizar una interfaz de 1,8V aun cuando la mayoría de

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

aplicaciones requieren una interfaz de 5V. ¿Cómo un aspirante a técnico de hardware puede resolver estos problemas?

No te preocupes, existe una solución: la placa protectora E/S. Esta placa se puede conectar a ODROID-U3 mediante un diminuto conector E/S de 8 pines. Tiene exactamente la misma forma y dimensiones que ODROID-U3, y viene con 12 separadores PCB para que su montaje sea sencillo y sólido.

La placa E/S tiene 2 bloques. “Bloque de expansión I2C E/S” y “bloque compatible con Arduino.”



Figura 4.63 Vista de la GPIO

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	



Figura 4.64 GPIO montada con la ODROID U3



Figura 4.65 Diagrama de Bloque comunicación Odroid & GPIO

Expansión I2C E/S: 16 x GPIO

Este bloque usa el extensor TCA6416A I2C a puerto paralelo.

El principal beneficio de este dispositivo es su amplio rango de VCC. Puede funcionar desde 1,65V a 5,5V en el lateral puerto-P y en el lateral SCL /SDA por

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

separado. Permite Expansión bidireccional con interpretación a nivel de tensión GPIO entre 1,8 V SCL/SDA (VCCD) y 5V Puerto (VCCP).

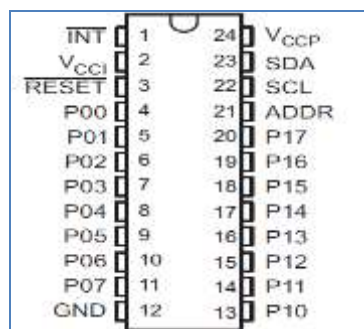



Figura 4.66 Expansión I2C

Los 16 puertos son asignados entre GPIO #289 ~ #304, que se puede leer desde la línea de comandos. Aquí hay un ejemplo para acceder a GPIO. Tenga en cuenta que necesitas para cargar el controlador primero el comando modprobe.

- # modprobe gpio-pca953x
- # echo tca6416 0x20 > /sys/devices/platform/i2c-gpio.4/i2c-4/new_device
- # echo 289 > /sys/class/gpio/export
- # cd /sys/class/gpio/gpio289
- /sys/class/gpio/gpio289# echo "in" > /sys/class/gpio/gpio289/direction
- /sys/class/gpio/gpio289# cat direction
- in/sys/class/gpio/gpio289# cat value

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Puede utilizar las librerías Python o C/C++ genéricas para acceder a los 16 GPIO en tu programa.

- **LIBRERÍA WIRING PI**

WiringPi es una librería de estilo Arduino escrita en c y desarrollada por Gordon Henderson, esta librería da un fácil acceso a los pines de propósito general del Raspberry Pi, la librería tiene soporte para el puerto uart, spi, i2c, pwm en la página del desarrollador se encuentra la información con mayor detalle, [link](#).


El proceso de instalación se realiza de forma muy sencilla siguiendo los siguientes pasos “logeado” como súper usuario:

- Instalar GIT : `sudo apt-get install git-core`
- Actualizar : `sudo apt-get update`
- Descargar wiringPi : `git clone git://git.drogon.net/wiringPi`
- Acceder a la carpeta descargada: `cd wiringPi` y luego `git pull origin`
- Acceder a la carpeta interna: `cd wiringPi`
- Instalar : `./build`

Ingresando a la carpeta wiringPi descargada, se encuentran ejemplos para probar las herramientas de la librería. el siguiente ejemplo realiza el on/off de un led con un delay de 500 ms.

- /*

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

- blink.c:
- Standard "blink" program in wiringPi. Blinks an LED connected
- to the first GPIO pin.
- *
- */
- #include <stdio.h>
- #include <wiringPi.h>
- // LED Pin - wiringPi pin 0 is BCM_GPIO 17.
- #define LED
- int main (void)
- {
- printf ("Raspberry Pi blink\n") ;
- if (wiringPiSetup () == -1)
- return 1 ;
- pinMode (LED, OUTPUT) ;
- for (;;)
- {
- digitalWrite (LED, 1) ; // On
- delay (500) ; // mS

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

- digitalWrite (LED, 0) ; // Off
- delay (500) ;
- }
- return 0 ;
- }

Aquí el diagrama de Conexión:

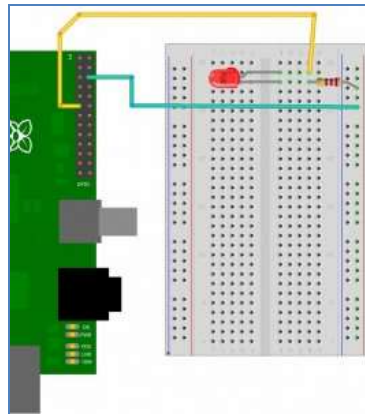



Figura 4.67 Diagrama de Conexión

El proceso de ejecución se realiza desde el terminal siguiendo los siguientes comandos:

- root@raspberrypi:/home/pi/wiringPi/examples# cc -o blink blink.c -L/usr/local/lib -lwiringPi
- root@raspberrypi:/home/pi/wiringPi/examples# make blink

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISIÓN 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 3
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

- `root@raspberrypi:/home/pi/wiringPi/examples# ./blink`

Siguiendo estos pasos se ejecuta la aplicación.

5. CONCLUSIONES.

Al terminar esta práctica el alumno de Ingeniería Electrónica debe establecer sus conclusiones.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 4
LABORATORIO	ELECTRONICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

PRÁCTICA No 4

NÚMERO DE ESTUDIANTES 20

ING. LUIS CÓRDOVA

TIEMPO ESTIMADO 2 HORAS

TEMA:

**CONFIGURACIÓN Y USO DE LOS PINES UARTS PARA LA
COMUNICACIÓN RS-232 EN EL PUERTO GPIO.**

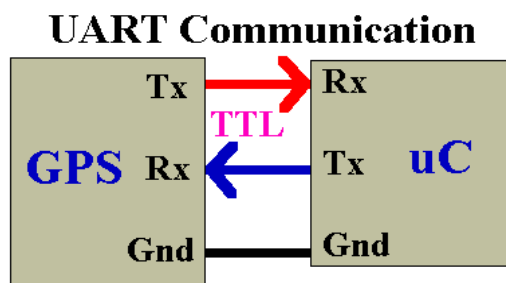



Figura 4.68 Comunicación Serie

Fuente: ElectroSome (2013). Recuperado de <https://electrosome.com/wp-content/uploads/2013/08/uart-communication.gif>

1. OBJETIVO GENERAL.

- Familiarizar al estudiante con las configuraciones que se realizan en el puerto UART para la comunicación vía puerto serial.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 4
LABORATORIO	ELECTRONICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

2. OBJETIVOS ESPECÍFICOS.

- Conectar correctamente los pines GPIO correspondientes a la UART del puerto serie.
- Conocer de manera clara los niveles de tensión presente en los pines de la UART y utilizar interfaces adecuadas para la comunicación RS-232.

3. MARCO TEÓRICO.

Conexiones y niveles de señal

El puerto serie Raspberry Pi se compone de dos señales (una señal de 'transmisión', TxD y una "recibir" señal RxD) disponible en el encabezado GPIO . Para conectarse a otro dispositivo serie, se conecta la 'transmisión' de uno a 'recibir' de la otra, y viceversa. También tendrá que conectar los pines de tierra de los dos dispositivos juntos.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 4
LABORATORIO	ELECTRONICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

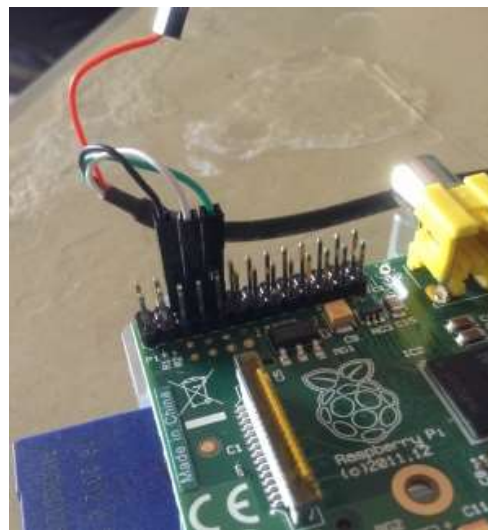


Figura 4.69 Conexión a los pines serie GPIO

El chip Broadcom en el corazón del Pi utiliza 0 y lógica 3.3V niveles, no el +/- 12V utilizado por RS-232 puertos serie que se encuentran en algunos equipos antiguos. Si desea conectar uno de estos, usted necesita un tablero o adaptador para convertir los niveles de señal.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 4
LABORATORIO	ELECTRONICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

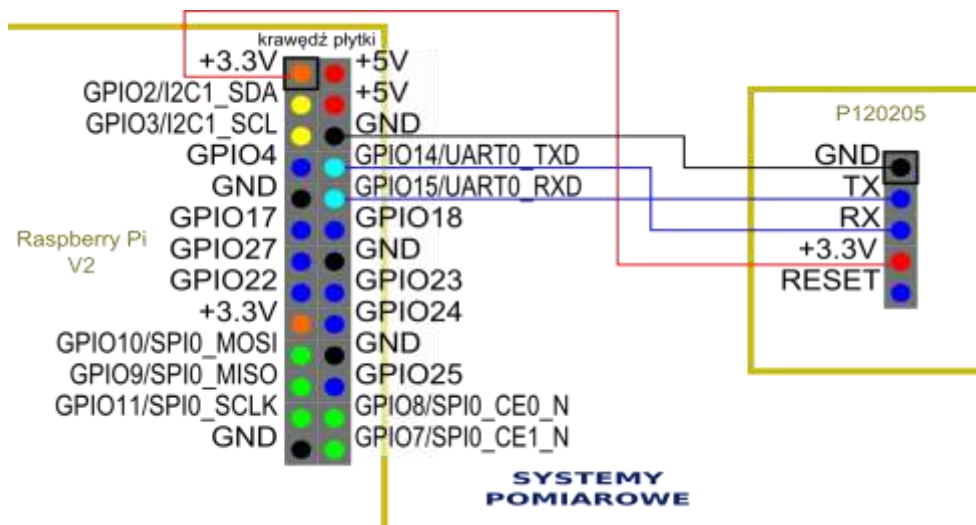


Figura 4.70 Pines GPIO correspondientes a la comunicación serie

Fuente: Systemypomiarowe (2013). Recuperado de http://www.systemypomiarowe.pl/wp-content/uploads/2013/04/P120205_poloczenie.png

Si desea conectar el Pi a un PC con un puerto USB, la opción más sencilla es utilizar un cable USB a serie que utiliza niveles lógicos 3.3V (por ejemplo los TTL-232R-RPI FTDI cables). Estos pueden ser simplemente conectados directamente a la cabecera GPIO (ver figura).

Si desea conectarse a un periférico que tiene 0 / 5V señales, lo ideal sería tener un circuito para convertir entre los niveles de tensión.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 4
LABORATORIO	ELECTRONICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Otros circuitos para cambio de nivel se muestran en RPi_GPIO_Interface_Circuits # Level_Shifters .

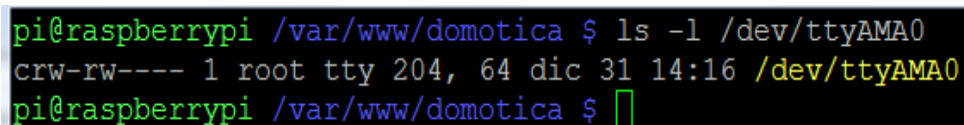
4. DESARROLLO.

Configuración de la UART

Para configurar la UART y poder usarla para conectar dispositivos tales como Arduino se debe hacer lo siguiente.

Este es el puerto serie:

```
ls -l /dev/ttyAMA0
```



```
pi@raspberrypi /var/www/domotica $ ls -l /dev/ttyAMA0
crw-rw---- 1 root tty 204, 64 dic 31 14:16 /dev/ttyAMA0
pi@raspberrypi /var/www/domotica $
```

Figura 4.71 Comando para verificación de accesibilidad

Es accesible para root y para los usuarios asociados al grupo tty.

Para asociar nuestro usuario a ese grupo:

- sudo usermod -a -G tty pi
- sudo reboot

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 4
LABORATORIO	ELECTRONICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Para ver el proceso que está corriendo en la RPi en el puerto serie.

ps ax | grep getty

Este proceso es para hacer login en el puerto serie y conectarse a él.

```

pi@raspberrypi /var/www/domotica $ ps ax | grep getty
2788 tty1      Ss+  0:00 /sbin/getty --noclear 38400 tty1
2789 tty2      Ss+  0:00 /sbin/getty 38400 tty2
2790 tty3      Ss+  0:00 /sbin/getty 38400 tty3
2791 tty4      Ss+  0:00 /sbin/getty 38400 tty4
2792 tty5      Ss+  0:00 /sbin/getty 38400 tty5
2793 tty6      Ss+  0:00 /sbin/getty 38400 tty6
2794 ?        Ss+  0:00 /sbin/getty -L ttyAMA0 9600 vt100
6340 pts/0    S+   0:00 grep --color=auto getty
pi@raspberrypi /var/www/domotica $ █

```

Figura 4.72 Comando para verificación de proceso.


Pero para poder controlarlo, hay que deshabilitar el proceso getty que está interfiriendo en la comunicación serie.

Si no hubiese salido nada al ejecutar el comando ps entonces los dos siguientes pasos se pueden obviar.

sudo nano /etc/inittab

Comentar la última línea con un # justo antes de T0:23

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 4
LABORATORIO	ELECTRONICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

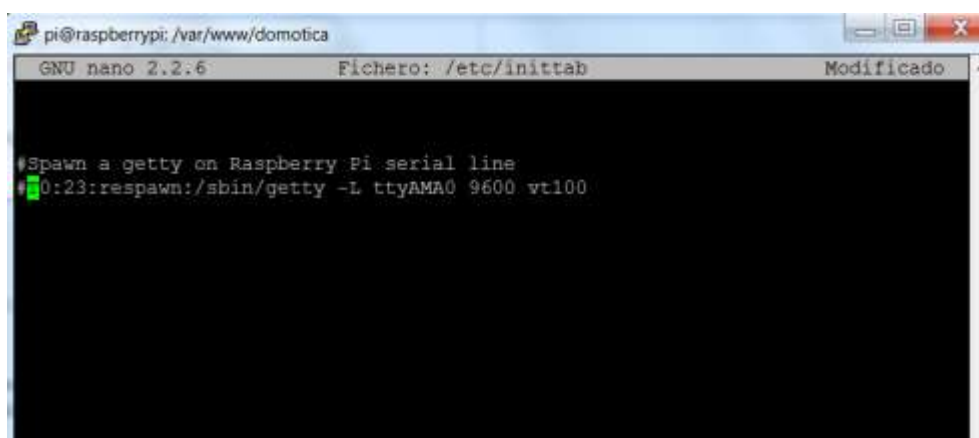


Figura 4.73 Línea a comentar en fichero /etc/inittab.

Comentar la última línea.

control +x y guardar.

Para que inittab realice los cambios oportunos.

sudo kill -1 1

Hasta aquí se podía obviar si no se tenía el proceso getty corriendo.

Lo siguiente es para que la salida del puerto serie no quede registrada en los ficheros log del sistema.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No 4
LABORATORIO	ELECTRONICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

sudo nano /boot/cmdline.txt

Quitar la siguiente porción de línea:

console=ttyAMA0,115200 kgdboc=ttyAMA0,115200

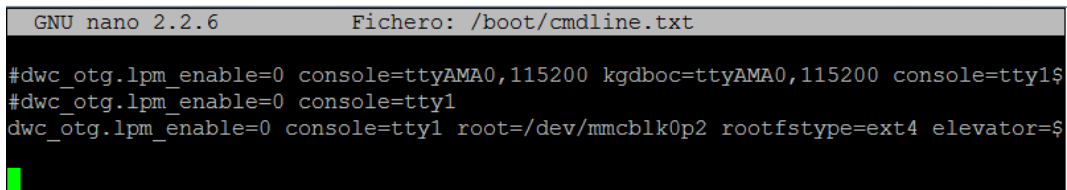


Figura 4.74 Edición de fichero /boot/cmdline.txt

Quitar console=ttyAMA0,115200 kgdboc=ttyAMA0,115200

Para reiniciar

sudo shutdown -r now

5. CONCLUSIONES.

Al terminar esta práctica el alumno de Ingeniería Electrónica debe estar en la capacidad de establecer sus propias conclusiones.

PRÁCTICA No 5

NÚMERO DE ESTUDIANTES 20

ING. LUIS CÓRDOVA

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 5
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

TIEMPO ESTIMADO 2 HORAS

TEMA:

INTRODUCCIÓN A PHYTON

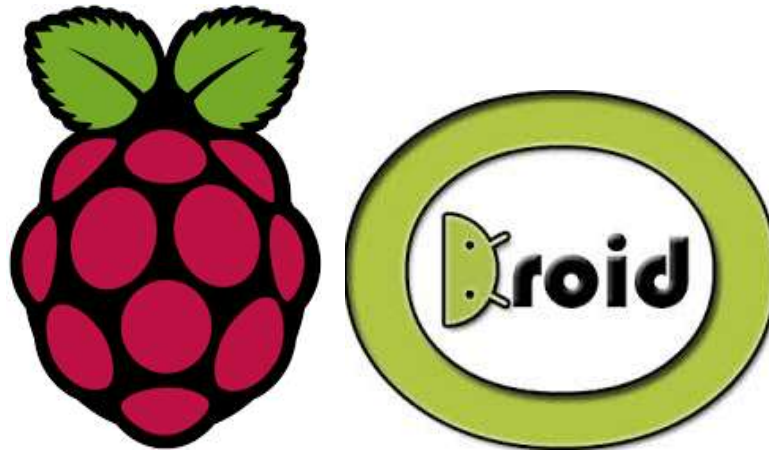


Figura 4.75 Logo de Equipos empleados

1. OBJETIVO GENERAL

- Conocer el lenguaje de programación PYTHON

2. OBJETIVOS ESPECIFICOS.

- Introducir al estudiante en conocimiento de nuevas técnicas de programación.
- Enseñar y aplicar la nueva sintáxis de programación.

3. MARCO TEÓRICO.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 5
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

¿Qué es Python?

Es el lenguaje de programación que es recomendado por los fundadores de la Raspberry Pi, pues piensan que es un lenguaje de sintaxis sencilla y clara que puede venir bien en los temas de educación.

Es un lenguaje interpretado o de script, fuertemente tipado y dinámico, es multiplataforma y es orientado a objetos.

Además, es un lenguaje bastante potente y con muchas librerías que ayudan a realizar casi cualquier cosa.

Python ya viene instalado de serie en las distribuciones que ofrece la fundación.

Existen 2 formas de ejecutar código en Python, una es escribiendo líneas en el intérprete de Python y obtener una respuesta a la vez, y la otra manera es escribir el código en un archivo de texto y ejecutarlo.

Probar con el ejemplo típico del Hola Mundo Interprete:

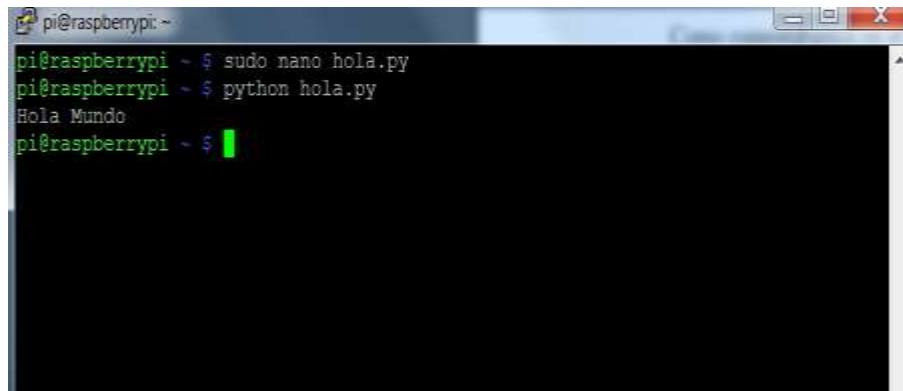
```
pi@raspberrypi ~ $ python
Python 2.7.3 (default, Jan 13 2013, 11:20:46)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print "Hola Mundo"
Hola Mundo
>>> |
```

Figura 4.76 Aplicación de Ejemplo

Código en archivo de texto:

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 5
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	



```

pi@raspberrypi ~ $ sudo nano hola.py
pi@raspberrypi ~ $ python hola.py
Hola Mundo
pi@raspberrypi ~ $ █

```

Figura 4.77 Ejecución de ejemplo

Para aprender a usar Python os recomiendo su página oficial <http://www.python.org/> y que leáis y practiquéis con alguno de los manuales/libros que existen.

4. DESARROLLO.

CONTROL DE UN LED

Seguir una serie de pasos para llevar a cabo esto. Seguro que así queda más claro.

PASO 1: MONTAJE

Antes de nada, realizar el montaje. Se necesita dos LEDs y un par de resistencias, por ejemplo de 1K Ohmios. Utilizar los GPIO 17 y 27. Quedará algo así:

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

REVISION 1/1	
	PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 5
LABORATORIO	ELECTRÓNICA-EMBEBIDOS
CARRERA	INGENIERÍA ELECTRÓNICA
SEDE	GUAYAQUIL

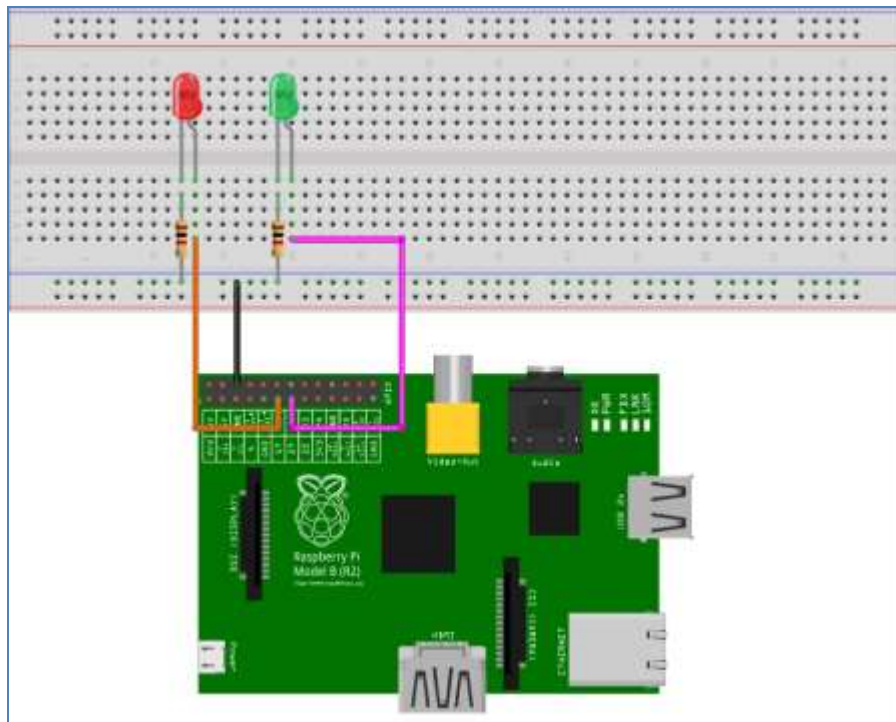


Figura 4.78 Conexionado en PROTOBOARD

PASO 2: INSTALACIÓN DE LA LIBRERÍA GPIO PARA PYTHON

Tras realizar el montaje, instalar la librería para poder controlar los GPIO con Python. Está alojada en SourceForge, descargarla en la Raspberry Pi con el siguiente comando:

- `Wget http://downloads.sourceforge.net/project/raspberry-gpio-python/RPi.GPIO-0.5.4.tar.gz`

Una vez descargada, descomprimir el tarball:

- `tar zxvf RPi.GPIO-0.5.4.tar.gz`

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 5
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Entrar en el directorio que se descomprime:

- `cd RPi.GPIO-0.5.4/`

Ahora toca instalar la librería. Por si no se tiene el paquete python-dev, introducir el siguiente comando:

- `sudo apt-get install python-dev`

Cuando acabe la instalación del paquete anterior, proceder a instalar la librería:

- `sudo python setup.py install`

PASO 3: PROGRAMA PYTHON PARA HACER UN ENCENDIDO INTERMITENTE DE LOS LEDS

En este tercer paso, escribir un pequeño programa en Python que haga que se enciendan y apaguen los LEDs de forma intermitente.

Crear un nuevo archivo Python.

- `sudo nano blink.py`

Importar la librería que se instala y declarar los pines.

Los pines serán de salida, se tiene que encender LEDs. Como ya se ha hecho en el montaje, utilizar los GPIO 17 y 27. La forma de tratar los pines, su declaración y el encendido y apagado os recordará a Arduino

- `import RPi.GPIO as GPIO`
- `import time`
- `GPIO.setmode(GPIO.BCM)`
- `GPIO.setup(17, GPIO.OUT) ## GPIO 17 como salida`

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

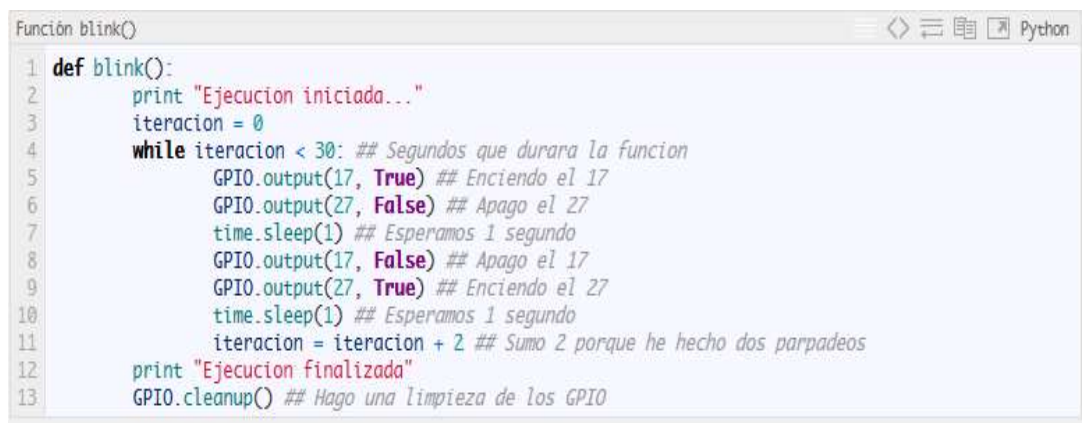
		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 5
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

- GPIO.setup(27, GPIO.OUT) ## GPIO 27 como salida

Crear una función para ejecutar el bucle que enciende y apaga los LEDs.

Por si no sabéis cómo definir funciones en Python. Aquí os dejo un tutorial:
Funciones en Python.

La función se llamará 'blink'. Es la típica con la que se empieza en Arduino.



```

Función blink()
1 def blink():
2     print "Ejecucion iniciada..."
3     iteracion = 0
4     while iteracion < 30: ## Segundos que durara la funcion
5         GPIO.output(17, True) ## Enciendo el 17
6         GPIO.output(27, False) ## Apago el 27
7         time.sleep(1) ## Esperamos 1 segundo
8         GPIO.output(17, False) ## Apago el 17
9         GPIO.output(27, True) ## Enciendo el 27
10        time.sleep(1) ## Esperamos 1 segundo
11        iteracion = iteracion + 2 ## Sumo 2 porque he hecho dos parpadeos
12    print "Ejecucion finalizada"
13    GPIO.cleanup() ## Hago una limpieza de los GPIO

```

Figura 4.79 Ejecución de los comandos

Llamar a la función.

- blink() ## Hago la llamada a la funcion blink

Con esto, ya se tiene todo el código hecho.

Ejecutamos el código.

- sudo python blink.py

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 5
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Ahora escribir el siguiente código en el editor de texto y guardar el archivo con el nombre hello.py.



Figura 4.80 Acción guardar archivo

```

1 import time
2 import RPi.GPIO as io
3 io.setmode(io.BCM)
4
5 pinLed = 14
6 io.setup(pinLed, io.OUT)
7
8 while True:
9     print('hola')
10    io.output(pinLed,io.HIGH)
11    time.sleep(1)
12    io.output(pinLed,io .LOW)
13    time.sleep(1)

```

Figura 4.81 Ejecución de comando

Asegurar que se guarde en la carpeta “pi”.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 5
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	



Figura 4.82 Guardar archivo

Abrir una ventana de terminal “LXTerminal” y ejecutar los siguientes comandos:

- `sudo python hello.py`

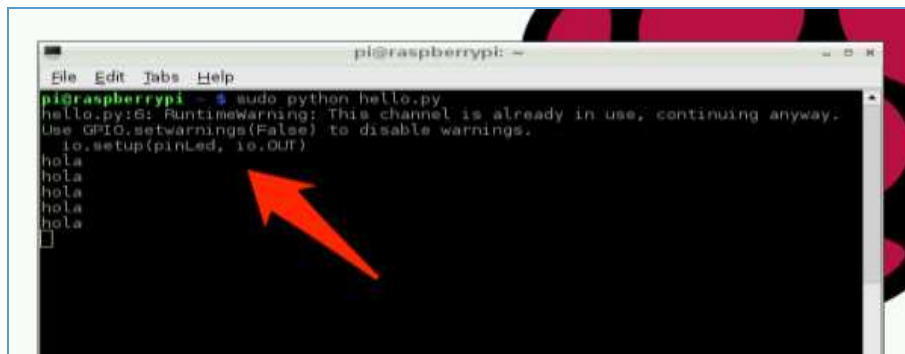


Figura 4.83 Prueba final líneas de comando

Y se ejecutará lo siguiente en la terminal. Además nuestro LED empezará a parpadear. Para poder detener el script presionar `ctrl+z`.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 5
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

5. CONCLUSIONES.

Al terminar esta práctica el alumno de Ingeniería Electrónica debe estar en la capacidad de establecer sus propias conclusiones.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 6
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

PRÁCTICA No 6

NÚMERO DE ESTUDIANTES: 20

ING. LUIS CÓRDOVA

TIEMPO ESTIMADO 2 HORAS

TEMA:

**CONTROL DE UN MOTOR DE PASOS Y DE UN SERVOMOTOR
PROGRAMANDO EN PYTHON.**



Figura 4.84 Logo PYTHON.

Fuente: Python Software Foundation (2015). Recuperado de <https://www.python.org/>

1. OBJETIVO GENERAL.

Familiarizar al estudiante con el uso del programa PYTHON en aplicaciones de control para los sistemas embebidos.

2. OBJETIVOS ESPECÍFICOS.

- Conocer en entorno de programación de PYTHON.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 6
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

- Familiarizar al estudiante con el manejo de los puertos GPIO por medio de PYTHON.
- Como utilizar PYTHON en la elaboración de programas para generar secuencias de control.

3. MARCO TEÓRICO.

¿Qué es PYTHON?

Python es un lenguaje que todo el mundo debería conocer. Su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo, muy rápido y, lo que es más importante, divertido.

La sintaxis de Python es tan sencilla y cercana al lenguaje natural que los programas elaborados en Python parecen pseudocódigo. Por este motivo se trata además de uno de los mejores lenguajes para comenzar a programar.

Python no es adecuado sin embargo para la programación de bajo nivel o para aplicaciones en las que el rendimiento sea crítico.

Algunos casos de éxito en el uso de Python son Google, Yahoo, la NASA, Industrias Light & Magic, y todas las distribuciones Linux, en las que Python cada vez representa un tanto por ciento mayor de los programas disponibles.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 6
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

4. DESARROLLO.

Después de haber conocido el manejo y uso de los puertos GPIO el siguiente paso importante es utilizar un motor paso a paso.

Interfaz con la Raspberry- Pi



Figura 4.85 Motor paso a paso

El motor se conecta a la tarjeta de interface con un conector pre-suministrado. La tarjeta de conexión tiene 4 + 2 pines que necesitan estar conectados a la cabecera Pi (P1).

- 5V (P1-02)
- GND (P1-06)
- Inp1 (P1-18)
- Inp2 (P1-22)
- Inp3 (P1-24)

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 6
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

- Inp4 (P1-26)

Las referencias P1-XX anteriormente representan las cabeceras de Pi pasadores que he usado. Estos se definen en el ejemplo de Python a continuación en la lista StepPins así que si usted utiliza diferentes pines asegúrese de actualizar la lista de Python también. Puede utilizar otros pines GPIO si se requiere sólo recuerde actualizar la secuencia de comandos de Python.

Para girar el motor pasó a paso que usted proporciona una secuencia de niveles "altos" y "bajos" a cada una de las 4 entradas en secuencia. Al establecer la secuencia correcta de niveles altos y bajos del husillo motor girará. La dirección puede ser revertida mediante la inversión de la secuencia.

Python script

Aquí está una copia de la escritura de motor paso a paso he usado para hacer girar el motor paso a paso. Utiliza la biblioteca RPi.GPIO y define una secuencia de 4 pasos y 8 pasos.

```
#!/usr/bin/env python
# Import required libraries
import sys
import time
import RPi.GPIO as GPIO

# Use BCM GPIO references
# instead of physical pin numbers
GPIO.setmode(GPIO.BCM)
```

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 6
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

```

# Define GPIO signals to use
# Physical pins 11, 15, 16,18
# GPIO17, GPIO22, GPIO23, GPIO24
StepPins = [17,22,23,24]

# Set all pins as output
for pin in StepPins:
    print "Setup pins"
    GPIO.setup(pin,GPIO.OUT)
    GPIO.output(pin, False)

# Define advanced sequence
# as shown in manufacturers datasheet
Seq = [[1,0,0,0],
        [1,1,0,0],
        [0,1,0,0],
        [0,1,1,0],
        [0,0,1,0],
        [0,0,1,1],
        [0,0,0,1],
        [1,0,0,1]]

StepCount = len(Seq)-1
StepDir = 2 # Set to 1 or 2 for clockwise
            # Set to -1 or -2 for anti-clockwise

# Read wait time from command line
if len(sys.argv)>1:
    WaitTime = int(sys.argv[1])/float(1000)
else:

```

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 6
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

WaitTime = 10/float(1000)

Initialise variables

StepCounter = 0

Start main loop

while True:

for pin in range(0, 4):

xpin = StepPins[pin]

print StepCounter

print pin

if Seq[StepCounter][pin]!=0:

print " Step %i Enable %i" %(StepCounter,xpin)

GPIO.output(xpin, True)

else:

GPIO.output(xpin, False)

StepCounter += StepDir

If we reach the end of the sequence

start again

if (StepCounter>=StepCount):

StepCounter = 0

if (StepCounter<0):

StepCounter = StepCount

Wait before moving on

time.sleep(WaitTime)

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 6
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

5. CONCLUSIONES.

Al terminar esta práctica el alumno de Ingeniería Electrónica debe estar en la capacidad de establecer sus propias conclusiones.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

PRÁCTICA No 7

NÚMERO DE ESTUDIANTES 20

ING. LUIS CÓRDOVA

TIEMPO ESTIMADO 2 HORAS

TEMA:

INTRODUCCIÓN A GNU Octave y Scilab/Xcos



Figura 4.86 Programas a utilizar

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

1. OBJETIVO GENERAL.

- Familiarizar al estudiante con el uso de los programas GNU Octave y Scilab/Xcos.

2. OBJETIVOS ESPECÍFICOS.

- Utilizar comandos básicos de GNU Octave y Scilab/Xcos aplicados a las técnicas de control.
- Fomentar y fortalecer el uso de software libre en el desarrollo y análisis de sistemas de control.
- Establecer comparaciones de las ventajas y alcances existentes entre los programas MATLAB, GNU Octave y Scilab/Xcos.

3. MARCO TEÓRICO.

GNU Octave

GNU Octave es un lenguaje de alto nivel interpretado, destinado principalmente para cálculos numéricos. Proporciona capacidades para la solución numérica de problemas lineales y no lineales, y para realizar otros experimentos numéricos. También proporciona amplias capacidades de gráficos para la visualización y manipulación de datos. Octave se utiliza normalmente a través de su interfaz de línea de comandos interactiva, pero también puede ser utilizado para escribir programas no interactivos. El lenguaje de Octave es bastante similar a Matlab para que la mayoría de los programas son fáciles de transportar.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Octave se distribuye bajo los términos de la GNU General Public License .

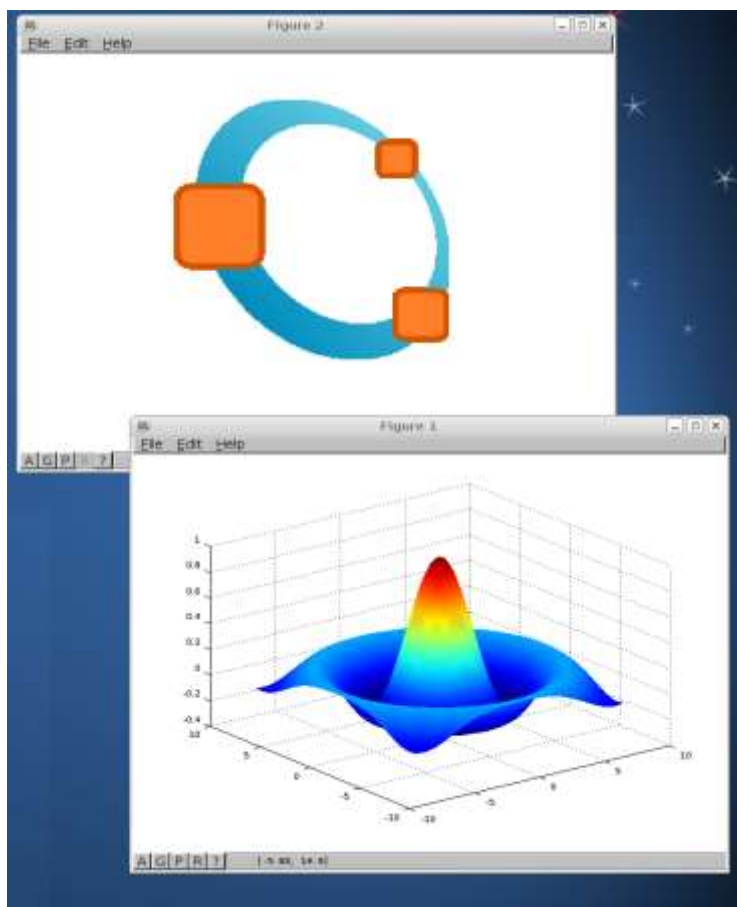


Figura 4.87 Captura de pantalla Octave

Scilab/Xcos

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Scilab es un software libre y de código abierto para el cálculo numérico que proporciona un entorno de computación de gran alcance para aplicaciones de ingeniería y científicas.

Scilab es liberado como código abierto bajo la licencia CeCILL (GPL compatible), y está disponible para su descarga gratuita. Scilab está disponible bajo GNU / Linux, Mac OS X y Windows XP / Vista / 7/8.

Scilab incluye cientos de funciones matemáticas. Tiene un lenguaje de programación de alto nivel que permita el acceso a estructuras de datos avanzadas, 2-D y funciones gráficas 3-D.

El programa Scilab tiene un entorno similar a Simulink de Matlab para simulación de sistemas dinámicos y resolución de sistemas de ecuaciones diferenciales. Este entorno posee varios paquetes que incluye algunas herramientas para simulación sencilla de circuitos eléctricos y termo hidráulica.

En el pasado Scilab podía ser utilizado en el análisis de sistemas, pero no podía interactuar con el exterior. Hoy en día se pueden construir interfaces para que desde Scilab se pueda manejar un dispositivo, se conecte a la red a través de Tcp (Protocolo de Control de Transmisión) o Udp (User Datagram Protocol), etc. Esto brinda la posibilidad de conectar una placa de adquisición de datos a Scilab y de esta forma el control de una planta on-line.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

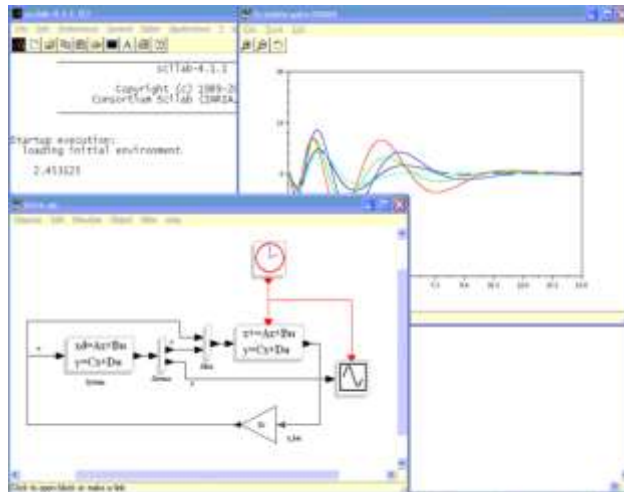


Figura 4.88 Pantallas de Scilab

4. DESARROLLO.

PARTE 1: GNU Octave.

Arranque de la consola de Octave

Partir de un ordenador con Octave instalado. Abrir un terminal para poder ejecutar instrucciones del sistema operativo. En la terminal de Linux (Raspberry-Pi && ODROID-U3) simplemente hay que teclear ‘*octave*’, en *Windows* el programa que ejecuta la consola de Octave es un programa llamado *ido* bien, aparece la consola de *Octave* con el *prompt* indicando algo parecido a *octave:1>*.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

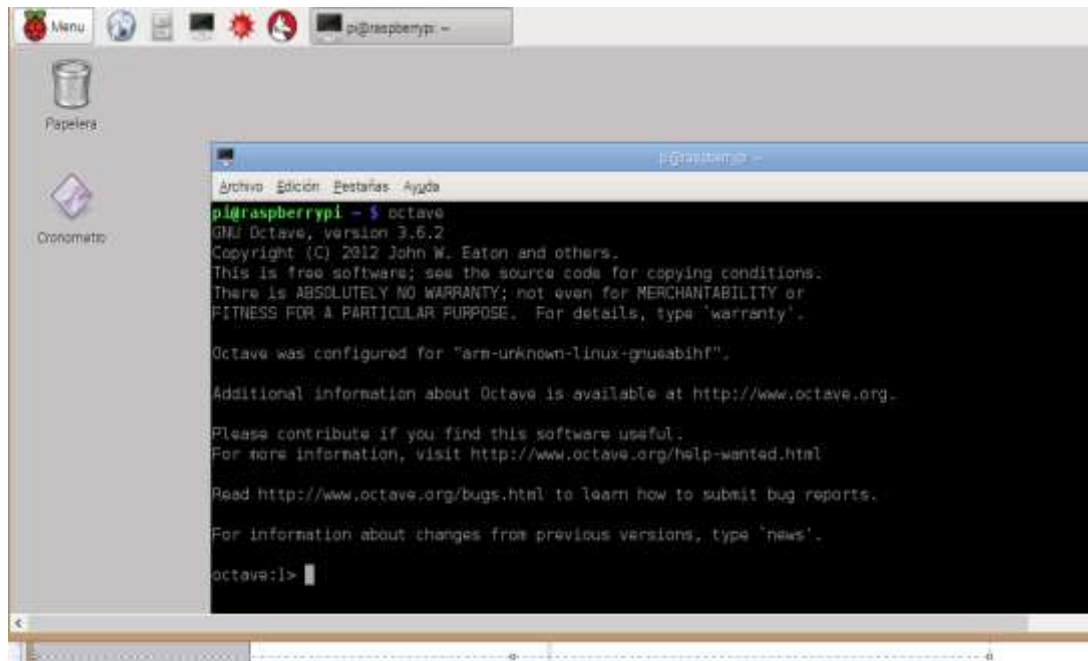


Figura 4.89 Terminal de Octave

Para salir de la consola de Octave hay que teclear ‘exit’ o ‘quit’

1. Cálculos elementales

Octave se puede utilizar como una calculadora. Se dispone de las operaciones aritméticas suma, resta, multiplicación, división y exponenciación. Pruebe a teclear las siguientes operaciones en la consola de Octave:

1+1

2-1

2*3.5

3.0/2.0

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

2⁵

Tras cada operación hay que pulsar la tecla *intro*. La consola muestra la palabra *ans*, el signo igual y el resultado de la operación. La palabra *ans* quiere decir *answer*, respuesta.

Octave dispone de funciones para calcular raíces cuadradas, logaritmos naturales, logaritmos decimales y funciones trigonométricas. Pruebe las siguientes operaciones en la consola de Octave:

`sqrt(25)`

`log(10)`

`log10(10)`

`sin(90*pi/180)`

`cos(0)`

`tan(45*pi/180)`

`asin(1)*180/pi`

`acos(0)`

`atan(-1)*180/pi`

Se observa un par de cosas en las expresiones anteriores:

- La utilización de la constante predefinida *pi*. Octave tiene varias constantes con valores predefinidos, una de ellas es el número $pi=3.14\dots$
- Las funciones trigonométricas trabajan con ángulos en radianes.

Se puede calcular exponenciales del número e , por ejemplo, la ecuación de *Euler*:

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Se comprueba tecleando algo así:

$$\exp(\pi*i)+1$$

Se observa que Octave entiende la variable i como el número complejo

Se puede ver algunas de las variables que tiene predefinidas:

e
i
j
pi
ans

Cada vez que hace un cálculo, el resultado se guarda en una variable llamada *ans* que se puede utilizar en el siguiente cálculo.

1. Asignación de variables

Se puede almacenar valores en memoria mediante la asignación de dichos valores a *nombres válidos* de variables. El símbolo utilizado para la asignación es el símbolo igual '='.

En el ejemplo siguiente se asigna valor a las variables x e y . Estos valores quedan almacenados en la memoria de *Octave* y se pueden utilizar en operaciones posteriores utilizando en las expresiones el nombre de variable elegido. Se puede reasignar el valor de una variable en cualquier momento. El valor en memoria será el último asignado a la variable.

$$x = 2.5$$

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

y = 3

x+y

% El resultado será: ans = 5.5

x = 4

x+y

% El resultado será: ans = 7

Se observa que el esquema de la *instrucción de asignación en *Octave* es situar el nombre de variable, a continuación el símbolo igual y a la derecha del símbolo igual el valor que se desea asignar a la variable.

A la derecha del símbolo igual puede aparecer un valor numérico tecleado explícitamente o cualquier expresión válida de *Octave*. En el siguiente ejemplo se asignan distintos valores a variables:

x = pi/2;

y = sin(x)

% El resultado será: y = 1

Note

La variable predefinida ans de *Octave* guarda el resultado de la última operación realizada en la consola de *Octave*, siempre que esa operación no sea de asignación. Si se realiza una asignación, el valor de la variable *ans* no varía.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Note

Si al hacer una asignación no se desea que muestre el resultado de la asignación en consola se tiene que finalizar la sentencia con punto y coma ‘;’.

2. Nombres de variables

Los nombres de variables pueden contener letras, números y caracteres underscore (guión bajo), pero el primer carácter tiene que ser letra.

Warning

No hay que utilizar como nombres de variables los nombres de variables predefinidas, funciones o comandos de *Octave*.

Note

Las letras que se pueden utilizar en los nombres de variables son las del alfabeto inglés. Los caracteres locales, (ñ, letras acentuadas), no se deben utilizar en el nombre de variables o funciones.

Ejemplos de nombres válidos de variables:

x

x2

XX_2

ultimoValor

n_factorial

Ejemplos de nombres no válidos de variables:

6x

end

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

n!

El número máximo de caracteres que puede tener el nombre de una variable se puede consultar con la función `namelengthmax()`:

```
namelengthmax()
```

```
% El resultado en mi consola: ans = 63
```

Note

Los nombres que dan una idea de para qué sirven las variables hacen que la legibilidad del código fuente de los programas mejore mucho. Es más fácil de seguir y comprender un programa, (un tercero o el mismo programador al cabo de unas semanas o meses), cuando los nombres de variables y funciones se eligen adecuadamente. Por ejemplo, la variable *numFilas* dice más que la variable *n*, y puede ser de gran ayuda para *seguir* el hilo del programa en una serie de bucles y sentencias *if* anidadas. Un criterio habitual es denominar a una variable con más de una palabra, poniendo la inicial de la primera palabra en minúsculas y las iniciales del resto de palabras en mayúsculas. Así se puede utilizar nombres de variables como *contadorVehiculos* o *ultimaFila*.

PARTE 2: Scilab/Xcos.

En esta sección se muestran los primeros pasos con Scilab y algunos ejemplos sencillos desde la consola.

Consola.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Este es el principal caso para manejar Scilab de una manera interactiva, por medio de escritura en el comando de la consola, analizando los resultados y continuando con su proceso hasta obtener resultados finales.

De esta manera se ejecutan varios comandos de las operaciones matemáticas tradicionalmente conocidas de una manera muy similar a la empleada en GNU Octave o MATLAB.



Figura 4.90 Ejecución de operaciones matemáticas Scilab.

Para mostrar un ejemplo de comandos se almacena en una variable y se volca su contenido:

```
a = "bienvenidos"
```

```
a
```

```
hola
```

```
disp(a)
```

```
hola
```

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

REVISION 1/1	
	PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS
CARRERA	INGENIERÍA ELECTRÓNICA
SEDE	GUAYAQUIL

3. Ayuda de comandos.

Para obtener ayuda con los comandos y ver una descripción de su operación simplemente se escribirá en la consola el comando help disp :

help disp



Figura 4.91 Ventana de ayuda del programa

Aparece el navegador de ayuda, junto con ejemplos de cómo usar el comando.

Los nombres de los comandos más utilizados son los siguientes:

sqrt(x)	Raíz cuadrada	sin(x)	Seno(radianes)
abs(x)	Módulo	cos(x)	Coseno(radianes)
Conj(z)	Complejo conjugado	Tan(z)	Tangente(radianes)
Real(z)	Parte real	Cotg(x)	Cotangente(radianes)
Imag(z)	Parte imaginaria	Asin(x)	Arcoseno
Exp(x)	Exponencial	Acos(x)	arcocoseno
Log(x)	Logaritmo natural	Atan(x)	Arcotangente
Log10(x)	Logaritmo decimal	Cosh(x)	Coseno hiperbolico
Rat(x)	Aprox racional	Sinh(x)	Seno hiperbolico
Modulo(x,y)	Resto dividir x por y	Tanh(x)	Tangente hiperbolica
Floor(x)	N tal que $n \leq x < (n+1)$	Acosh(x)	Arcoseno hiperbolico

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Figura 4.92 Lista de comandos más utilizados

Editor de Scripts.

Se puede utilizar un editor de scripts que le permite escribir todo un conjunto de comandos para luego ejecutarlos de manera continua, para acceder a este editor de scripts existen dos vías, la primera es desde la consola escribiendo:

- editor

y la segunda es por medio del menú applications/SciNotes.

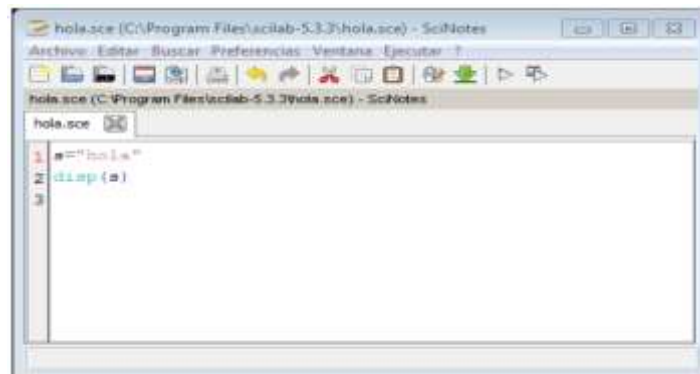


Figura 4.93 Editor de Scripts

Al ejecutar los programas del script, el resultado se muestra en la consola de Scilab.

Es posible integrar el editor a Scilab como se muestra en la figura a continuación:

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

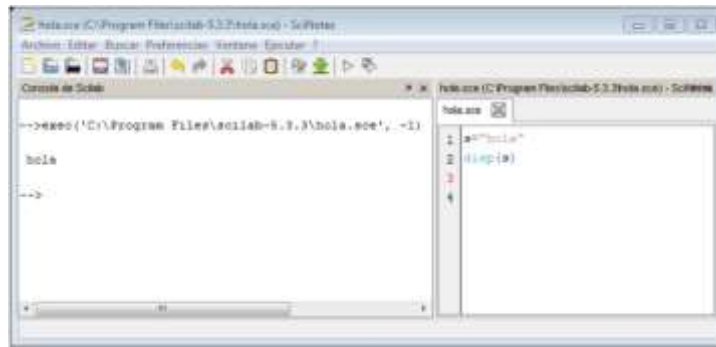


Figura 4.94 Integración del Editor en Scilab

Para colocar un comentario simplemente ubicar // antes de la línea a ser comentada.

- Introducción a Xcos

Parte de este documento es una traducción de la página de ayuda de Scilab, <http://help.scilab.org/>, que explica el funcionamiento del entorno de Xcos. Este documento pretende ayudar a los usuarios hispano hablantes a utilizar el programa, conociendo en primer lugar las opciones y características que este ofrece.

1. INTRODUCCIÓN

Scilab tiene un entorno similar a Simulink de Matlab para simulación de sistemas dinámicos y resolución de sistemas de ecuaciones diferenciales. Este entorno posee varios paquetes que incluye algunas herramientas para simulación sencilla de circuitos eléctricos, sistemas de control, termo hidráulica, etc. A este entorno de programación gráfica se le llama Xcos. No siempre existe la posibilidad de tener acceso a los programas de pago en el entorno de trabajo, es por ello, que es

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

importante aprender a utilizar los programas gratuitos que se pueden encontrar con similares características a las versiones de pago.

2. El Entorno de Xcos

Tras abrir Scilab, se puede acceder al Xcos bien escribiendo la sentencia xcos en el entorno de escritura de Scilab y pulsando ‘enter’, o bien pulsando sobre el icono que se ve en la figura siguiente.

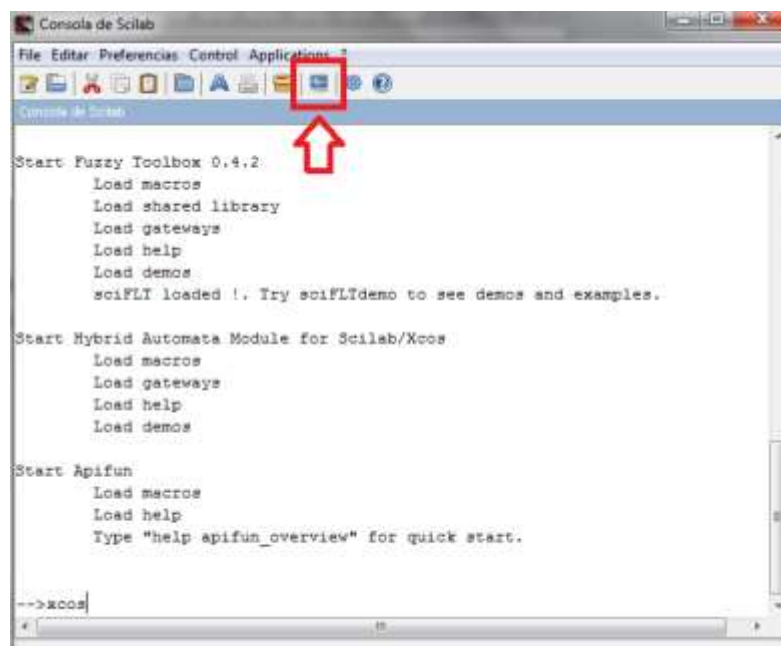


Figura 4.95 Ingreso a Xcos

1. Descripción de las ventanas de Xcos.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Xcos es un editor gráfico para la construcción de modelos de sistemas dinámicos híbridos. Los modelos se pueden construir, cargar, guardar, compilar, simular, mediante la GUI de Xcos. El entorno Xcos presenta los siguientes elementos: Editor. El editor permite el diseño de diagramas de flujo que representan a un sistema dinámico de bloques definidos en las paletas (en Simulink librerías). Los usuarios pueden personalizar los bloques estándar y definir otros nuevos.

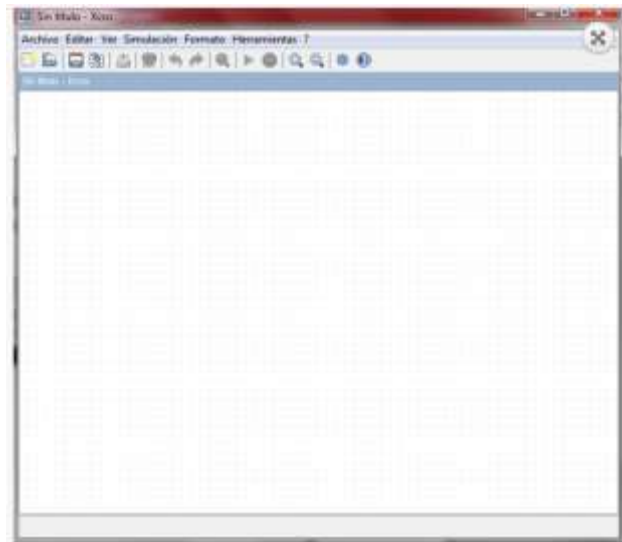


Figura 4.96 Ventana principal de Xcos

Explorador de paletas. El explorador de paletas contiene una lista donde se enumeran todos los bloques de Xcos estándar agrupados por categorías (procesamiento de señal, eléctrica, hidráulica, derivada, integral, etc.) Véase la siguiente figura.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

REVISION 1/1	
	PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS
CARRERA	INGENIERÍA ELECTRÓNICA
SEDE	GUAYAQUIL

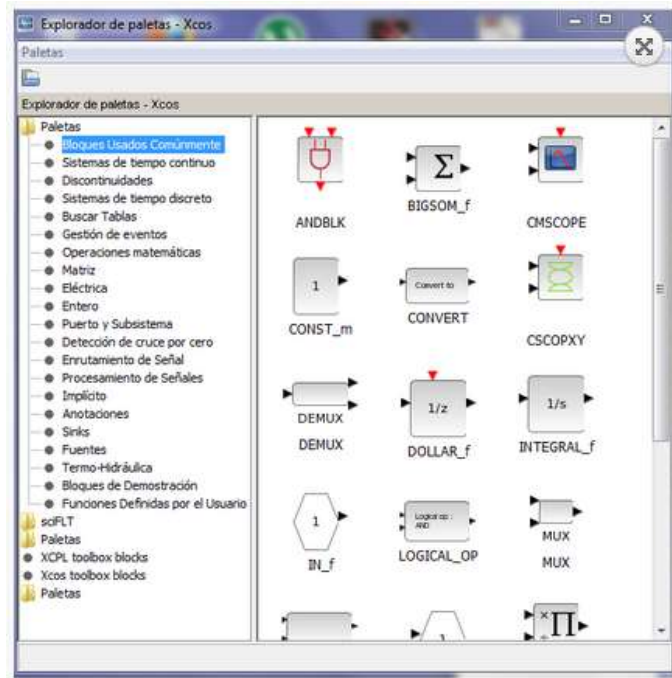


Figura 4.97 Explorador de Paletas

Compilador / Simulador El entorno de Xcos permite, a partir del modelo creado, la simulación de sistemas complejos y sistemas híbridos. Los datos resultantes de la simulación gráfica se pueden ver en tiempo real, utilizado en Scilab para el procesamiento posterior. Para abrir un modelo ya creado de Xcos, puedes escribir el nombre del archivo, como se puede ver en la siguiente sentencia: [crayon-54f375f9580cc523314243/] [crayon-54f375f9580de841171653/] También puedes acceder al él desde el menú de archivo del explorador de paletas de Xcos.

Opciones de edición en Xcos (Menú Simulación)

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

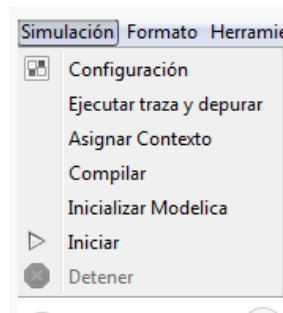


Figura 4.98 Ventana Simulación

Simulación: Configuración

En la ventana principal de Xcos, pulsando sobre el ítem Configuración, se abre una ventana de diálogo que permite cambiar los parámetros de integración:

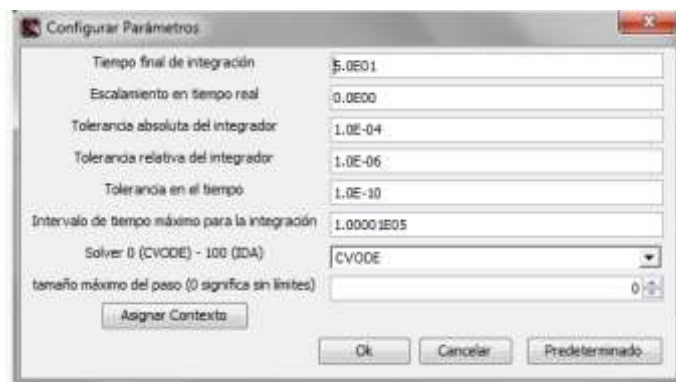


Figura 4.99 Ventana de configuración de la simulación.

Tiempo final de integración: la integración (simulación) termina en este momento, siempre se empieza desde 0

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Escalamiento en tiempo real: fuerza la simulación en tiempo real mediante el establecimiento de simulación en Xcos en una unidad de tiempo de 1 segundo.

Tolerancias absolutos del integrador: propiedades Solver.

La tolerancia relativa de tiempo: el intervalo de tiempo más pequeño para el cual se utiliza el solucionador de “ode” que actualiza los estados continuos.

Intervalo de tiempo máximo para la integración: el intervalo de tiempo máximo para cada llamada a solver.it debe ser reducido si se produce el mensaje de error "Demasiadas llamadas".

Solver: seleccionar la solución numérica para ser utilizada. Tienes la posibilidad de elegir entre una ecuaciones diferenciales ordinarias (ODE) Solver y una solucionador de ecuaciones diferenciales algebraicas (AIF). Si Xcos detecta que el modelo requiere un solucionador de la AIF, el Xcos muestreará un cuadro de información sobre el cambio automático al solucionador AIF.

Tamaño máximo de paso: el paso máximo de tiempo que tarda el solucionador. Este parámetro es útil para buscar singularidades localizadas en una respuesta del sistema monótono. Para este tipo de respuesta, la solución numérica aumenta automáticamente el tamaño de paso para disminuir el tiempo de cálculo. Si piensas que su sistema está en este caso, disminuye el valor de este parámetro.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

7.1 Ejemplo configuración muestreo CSCOPE

Se muestra la configuración del clock  usado en el CSCOPE, en el cual se escoge el número de muestra de nuestra señal. Se toma como referencia un generador senoidal y se grafica su salida, como se ve en la figura 24, para variar el periodo de muestreo se hace doble click en el  y aparece la ventana de la gráfica 25.



Figura 24 Generador senoidal

Se configura el periodo de muestreo en 0.1 para ver la señal del generador y se variara aumentándolo a 0.5 y 1 y se observa el comportamiento



Figura 25 Periodo muestreo CSCOPE

Figura 4.100 Ejemplo de configuración del muestreo

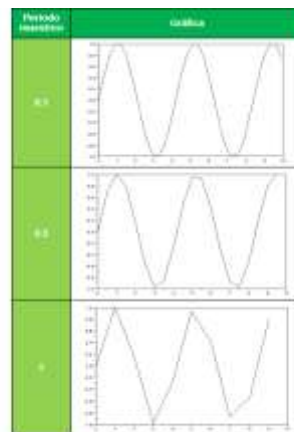


Tabla 2 periodo de muestreo
Como se observa entre más grande sea el tiempo de muestreo más perdida de onda va a existir, por lo cual se recomienda tomar muestras bajas para obtener una óptima señal.

Figura 4.101 Muestreo en ejecución

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

- RECURSOS UTILIZADOS
- Computadores
- MATLAB
- Guía de la practica
- REGISTRO DE RESULTADOS

¿Qué cambios ha sufrido la salida con cada uno de los tres tipos de entrada, al modificar la función de Transferencia, en el sistema de lazo abierto?

¿Qué cambios ha sufrido la salida con cada uno de los tres tipos de entrada, al modificar la función de Transferencia, en el sistema de lazo abierto?

¿Qué cambios ha sufrido la señal de salida para cada uno de los tres tipos de entrada, al incrementar un lazo de retroalimentación?

¿Explique desde su perspectiva cuales son las diferencias entre el sistema de lazo abierto y el de lazo cerrado?

¿Qué cambios se han dado en las señales de salida?

¿Cómo afecta la perturbación a la salida en cada uno de los sistemas de control?

- ANEXOS

Principales bloques utilizados en Xcos (Sistemas de tiempo continuo)

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

REVISION 1/1	
	
PROCEDIMIENTO PARA EL DESARROLLO PRÁCTICA No. 7	
LABORATORIO	ELECTRÓNICA-EMBEBIDOS
CARRERA	INGENIERÍA ELECTRÓNICA
SEDE	GUAYAQUIL

Aunque los principales bloques que se utilizarán en Xcos son en esencia muy similares a los de Simulink, existen algunas diferencias en el uso de estos que merece la pena comentar.

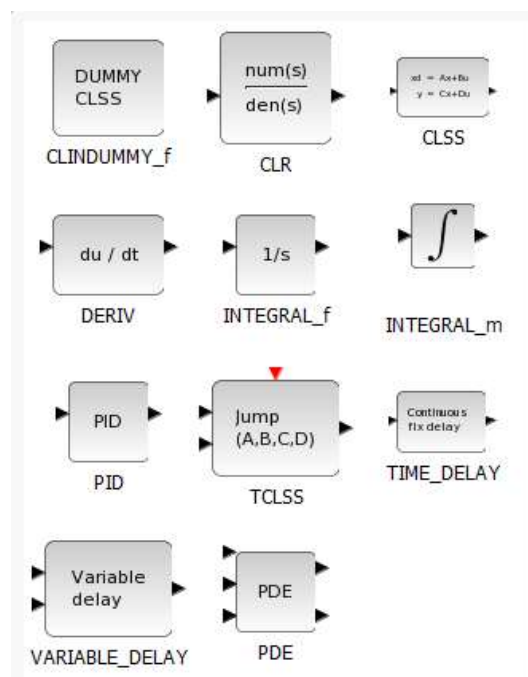


Figura 4.102 Bloques de integración

5. CONCLUSIONES.

Al terminar esta práctica el alumno de Ingeniería Electrónica debe estar en la capacidad de establecer sus propias conclusiones.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por:
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

PRÁCTICA No 8

NÚMERO DE ESTUDIANTES: 20

ING. LUIS CÓRDOVA

TIEMPO ESTIMADO 2 HORAS

TEMA:

INTRODUCCIÓN A QT CREATOR Y DESARROLLO DE INTERFACES GRÁFICAS EMPLEADAS EN INSTRUMENTACIÓN VIRTUAL Y COMPARACIÓN CON LAB-VIEW.



Figura 4.103 Logo programa QT CREATOR

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

1. OBJETIVO GENERAL

- Familiarizar al estudiante con el uso del programa QT-Creator.

2. OBJETIVOS ESPECÍFICOS.

- Promover y aplicar técnicas de Programación en QT.
- Fomentar y fortalecer el uso de software libre en el desarrollo de sistemas de monitoreo e instrumentación virtual.
- Conocer las prestaciones y alcances que ofrece Qt-Creator para aplicaciones de monitoreo e instrumentación industrial.

3. MARCO TEÓRICO.

Qt Creator

Es un Entorno Integrado de Desarrollo o IDE (editor + compilador + depurador) bastante completo, moderno, potente, fácil de manejar, eficiente, abierto y gratuito, que permite el desarrollo rápido de aplicaciones en entornos MS Windows, Mac OS y Linux. Algunos ejemplos de programas creados con las librerías Qt: Adobe Photoshop Album, Google Earth, KDE, Opera, Skype, VLC media player...etc.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Características fundamentales de Qt Creator:

- Utiliza el lenguaje de programación orientado a objetos C++.
- Se basa en Qt, una librería multiplataforma y gratuita para la creación de interfaces gráficas, programación web, multihilo, bases de datos, etc.
- Permite realizar programación visual y programación dirigida por eventos.
- Características avanzadas de IDE: sintaxis coloreada, compleción automática de código, ayuda sensible al contexto, inspector de objetos, diseñador visual, compilador y depurador integrado, etc.

Programación visual: el programador centra su atención en diseñar el aspecto gráfico de la aplicación, la distribución de los elementos visuales (llamados *widgets*: formularios, botones, menús, cuadros de texto, etc.), la interacción entre los mismos, los distintos tipos de ventanas existentes, etc.

Un entorno de programación visual se asemeja a un programa de dibujo, donde la imagen es una ventana (o formulario), y los elementos para dibujar son botones, etiquetas de texto, menús, etc.

El programador diseña el aspecto gráfico que tendrá la aplicación.

Programación dirigida por eventos: el programador escribe el código que se ejecutará en respuesta a determinados eventos (llamados *slots*: pulsar un botón, elegir una opción del menú, abrir o cerrar una ventana, etc.).

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

REVISION 1/1	
	
PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8	
LABORATORIO	ELECTRÓNICA-EMBEBIDOS
CARRERA	INGENIERÍA ELECTRÓNICA
SEDE	GUAYAQUIL

No existe la idea de un control de flujo *secuencial* en el programa, sino que el programador toma el control cuando se dispara un evento.

La labor del programador es asociar a cada evento el comportamiento adecuado.

Las ventanas son clases, los componentes (*widgets*) son clases, y los eventos (*slots*) son métodos de las ventanas. Nuestra ventana es una subclase de la clase ventana (QMainWindow, QDialog o QWidget).

1. EL ENTORNO DE QT CREATOR

Al ejecutar Qt Creator. Se abre una ventana como la que se muestra abajo.



Figura 4.104 Entorno QT Creator.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Modos de visualización.- Existen siete modos de visualización diferentes, que le permiten ver la información más adecuada en cada momento, según este editando, depurando, ejecutando, buscando ayuda, etc.

- **Modo Welcome (Bienvenida).**- Aparece siempre al empezar. Contiene tutoriales de Qt, ejemplos, y permite abrir proyectos rápidamente.
- **Modo Edit (Edición).**- Sirve para editar el código fuente de nuestra aplicación. Es el modo más habitual de visualización cuando está escribiendo el programa.
- **Modo Design (Diseño).**- Permite ver y modificar el diseño de las ventanas y formularios de nuestra aplicación.
- **Modo Debug (Depuración).**- Lo usa cuando este depurando la aplicación. Muestra el código y la información de depuración.
- **Modo Projects (Proyectos).**- Permite ver y configurar las opciones del proyecto. Normalmente no necesita tocarlo mucho.
- **Modo Analyze (Análisis).**- Sirve para medir el tiempo consumido en distintas operaciones del programa.
- **Modo Help (Ayuda).**- Muestra la ayuda de Qt. Desafortunadamente, no incluye ayuda de C/C++ ni de las STL. Se puede usar Internet:

<http://www.manpagez.com> Páginas MAN online (para C).

<http://www.cplusplus.com> Referencia completa de C++ y las STL.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Método de trabajo.- El proceso normal de trabajo empezará con la creación de un nuevo proyecto. Después diseñar el aspecto gráfico de la ventana (o ventanas) de nuestra aplicación. Escribir el código usando el modo Edición.

4. DESARROLLO.

NUESTRO PRIMER PROGRAMA QT

Para crear nuestro primer programa con Qt Creator y a analizar los ficheros de los que consta un proyecto.

- Dentro de Qt Creator, si tiene abierto algún proyecto lo cierra con: File | Close All Projects and Editors. A continuación realizar: File | New File or Project... (o también, en el modo Welcome, solapa Develop, botón Create Project).
- Dentro de la ventana New Project elegir en Projects la entrada Qt Gui Application. Clickear en Choose.
- En la siguiente ventana, dentro de Name: poner el nombre de nuestro proyecto. Por ejemplo: HolaMundoQt (sin espacios). En Create in: seleccionar el directorio base (bajo ese directorio se creará un subdirectorio HolaMundoQt con todos los ficheros del proyecto). Clickear en Next.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

- Seguidamente aparece otra ventana, Kit Selection, para elegir el kit con el que se quiere compilar el programa. El kit se refiere al compilador usado, opciones de compilación, entorno de destino, y modos de compilación (Debug y Release). Por defecto solo tendrá la opción de elegir el kit Desktop Qt 5.1.0 MinGW 32 bit.
- A continuación viene otra ventana, solicitando el nombre de la ventana principal de la aplicación (nombre de la clase y nombre de los ficheros asociados .h, .cpp y .ui), y el tipo de ventana: QMainWindow, QWidget o QDialog.

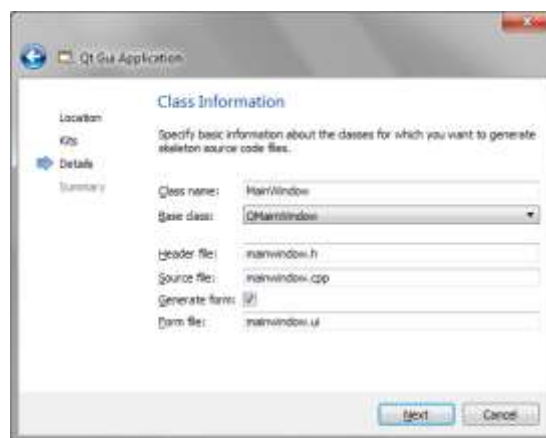


Figura 4.105 Ventana principal de la Aplicación

QMainWindow es una ventana con menú superior y barra de estado debajo. QWidget es una ventana vacía, con botones de maximizar, minimizar y cerrar. QDialog es una

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

ventana con botones de cerrar y ayuda. Dejar QMainWindow, dejar los nombres que están y clicar en Next.

Finalmente se da click en Finish y se genera el proyecto. ¡Ya se ha creado el primer proyecto Qt! Ahora observar la estructura, el contenido y el significado de los ficheros de un proyecto.

ESTRUCTURA DE UN PROYECTO QT

Dentro de Qt Creator seleccionar el modo Edit para ver los ficheros creados en el proyecto. Debe aparecer algo parecido a lo siguiente (después de desplegar las carpetas Headers, Sources y Forms).

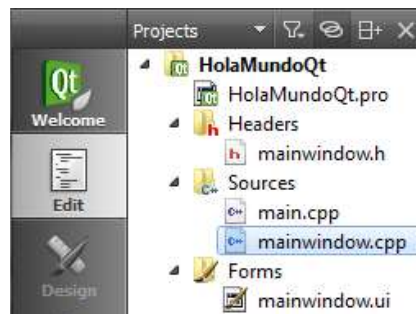


Figura 4.106 Estructura en QT CREATOR

Archivo: HolaMundoQt.pro Contiene la definición de todos los ficheros y elementos de los que se compone el proyecto. Es un archivo de texto. En concreto,

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

define el nombre de los formularios del proyecto (FORMS), de los ficheros de cabecera (HEADERS) y de implementación (SOURCES). También puede contener otra información, como la ruta de los ficheros *include* (INCLUDEPATH), las librerías externas utilizadas (LIB) y los ficheros de recursos (RESOURCES).

Directorio: Forms Dentro de este directorio se incluyen todos los formularios (ventanas) que contiene nuestra aplicación. La aplicación puede contener varias ventanas, pero solo una de ellas puede ser la ventana principal (la que se muestra al ejecutar el programa).

Archivo: mainwindow.ui Es la ventana principal de la aplicación. Las ventanas tienen siempre extensión .ui. Internamente son archivos de texto en formato XML, que describen los elementos visuales de la ventana (botones, etiquetas, menús, etc.). No editar “manualmente” sino de forma visual usando el editor de ventanas de Qt Creator (modo Design).

Directorio: Headers Dentro de este directorio están todos los archivos de cabecera (es decir, los .h). Recordar que los archivos de cabecera contienen la definición de las clases, constantes, variables y procedimientos públicos. En general tendrá dos tipos de ficheros de cabecera: asociados a las ventanas, o asociados a módulos no visuales.

Archivo: mainwindow.h Contiene la declaración de la ventana principal de nuestra aplicación. Nuestras ventanas serán siempre clases, declaradas como subclases de

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

QMainWindow, QWidget o QDialog. Observar el contenido del fichero mainwindow.h.

El archivo mainwindow.ui (que no es código C/C++ sino XML) se compila para generar automáticamente el fichero ui_mainwindow.h, donde se define una clase Ui::MainWindow, que es la que contiene los botones, etiquetas, etc. Después, dentro de mainwindow.h se define la clase MainWindow que incluye un atributo Ui::MainWindow *ui;

Directorio: Sources Dentro de este directorio está la implementación de las clases y de las funciones del programa. Contiene archivos de código C++ (con extensión .cpp). Pueden ser los asociados a las ventanas o asociados a módulos no visuales.

Archivo: mainwindow.cpp Aquí es donde va la implementación de los *slots* (los métodos asociados a los eventos de las ventanas), así como de cualquier otro método que se quiera añadir a la clase. Contiene algo de código generado automáticamente. También se puede definir procedimientos fuera de las clases, si lo necesitara.

Archivo: main.cpp Es el programa principal de nuestra aplicación, el procedimiento main. ¿Y qué hace el main?

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

```

main.cpp
1  #include "mainwindow.h"
2  #include <QApplication>
3
4  int main(int argc, char *argv[])
5  {
6      QApplication a(argc, argv);
7      MainWindow w;
8      w.show();
9      return a.exec();
10 }

```

Figura 4.107 Main.cpp

Se define una QApplication a, se define una MainWindow w, se muestra la ventana w y se ejecuta la aplicación a. Sencillo, ¿verdad?

20. Resumen. Se puede crear un nuevo proyecto con **Welcome | New File or Project... | Qt Gui Application**. El proyecto tiene un fichero de proyecto (.pro), un programa principal (main.cpp), una o varias ventanas, y posiblemente módulos adicionales. Las ventanas constan de un fichero con el formulario de la ventana (.ui), un fichero de cabecera (.h) y uno de implementación (.cpp). Los módulos no visuales contendrán el fichero de cabecera (.h) y el de implementación (.cpp).

¡HOLA MUNDO!

Hacer un sencillo programa “¡Hola Mundo!” y luego implementar algo un poco más avanzado.

Partir del proyecto HolaMundoQt que se ha creado en los pasos del 1 al 6. Seleccionar el modo Edit (en los botones de modos de visualización). En la lista de

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

REVISION 1/1	
	PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS
CARRERA	INGENIERÍA ELECTRÓNICA
SEDE	GUAYAQUIL

archivos del proyecto que aparece a la izquierda, seleccionar HolaMundoQt | Forms | mainwindow.ui y se hace doble clic encima.

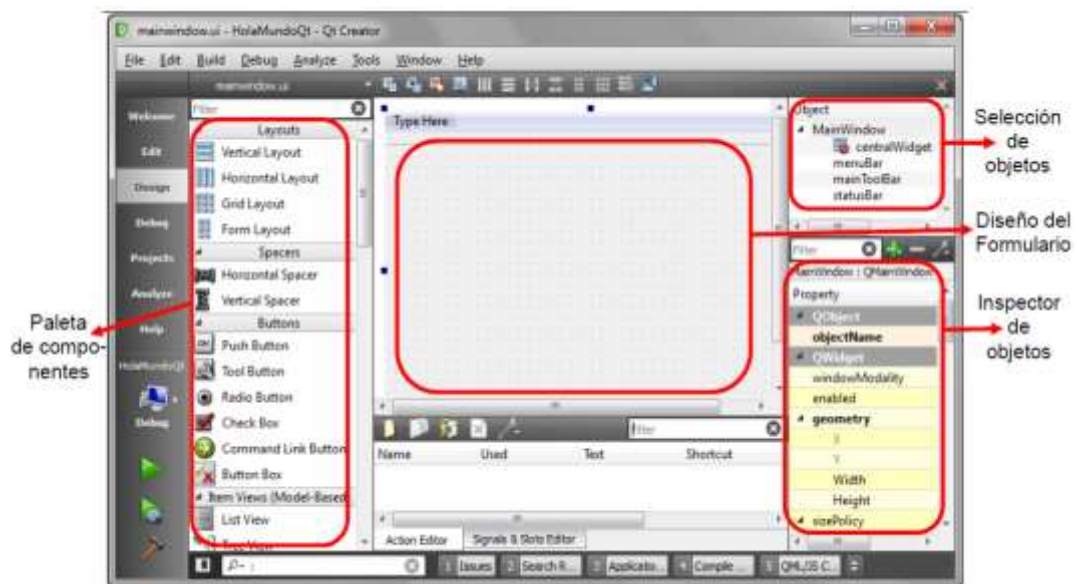


Figura 4.108 Pantalla Principal QT CREATOR

Se abre el editor de formularios (como se muestra arriba). A la izquierda aparece la paleta de componentes, con todos los elementos (*widgets*) que puede añadir a nuestro formulario. En el centro aparece el diseño del formulario con el que se trabaja. Y a la derecha aparece el inspector de objetos, donde puede editar las propiedades de los objetos que contiene el formulario.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Dentro del inspector de objetos, dirigir a la propiedad windowTitle y escribir: Mi primer programa Qt.

- A continuación, en la paleta de componentes, dar click con el ratón en el componente llamado Push Button y lo arrastra dentro del formulario. Puede cambiar su tamaño y posición según se desee; y también el texto. Para ello puede usar el inspector de objetos (propiedad texto), o bien hacer doble clic en el botón. Escriba: Saludar.
- Ahora escriba algo de código asociado al botón “Saludar”. Dar clic con el botón derecho del ratón sobre el botón “Saludar”. En el menú desplegable que aparece, seleccionar Go to slot... aparecerá una ventana con la lista de eventos que se pueden asociar al botón. Seleccionar cliché () y pulsar OK.
- Se abre automáticamente el fichero mainwindow.cpp dentro del editor de código, donde se crea un nuevo método de la ventana, asociado a la pulsación del botón “Saludar”. El cursor se queda esperando a que escriba el código del método.

```
void Main Window::on_pushButton_clicked()
{
}

```


- Dentro de las llaves escribir:

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

QMessageBox::information(this, "Mensaje", "Hola Mundo");

- Dentro del menú principal de Qt Creator, guardar el proyecto con File | Save All. En general, siempre es aconsejable grabar con cierta regularidad.

- Ahora ejecutar el programa con Build | Run, o bien con el botón 
 - Y el resultado es... ¡Uppps! Error de compilación: *'QMessageBox' has not been declared*. Claro, QMessageBox es una clase que está definida en la librería QMessageBox. En general, todas las clases de Qt están definidas en librerías que tienen el mismo nombre que la clase. En definitiva, hay añadir al principio:

```
#include <QMessageBox>
```

- De nuevo, guardar y ejecutar. Ahora sí, ¡ya ha creado el primer Hola mundo con Qt! Acuérdesse de cerrarlo antes de seguir.
- La ejecución conlleva implícitamente la compilación del proyecto (compilación + enlace). También se puede hacer explícitamente con Build | Build All.
- Observar los ficheros que se han creado dentro del directorio del proyecto:

HolaMundoQt.pro: fichero principal del proyecto.

main.cpp: programa principal de la aplicación.

mainwindow.ui, mainwindow.h, mainwindow.cpp: ficheros asociados a la ventana principal.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Y en el directorio: build-HolaMundoQt-Desktop_Qt_5_1_0_MinGW_32bit-Debug se tiene el resultado de la compilación:

ui_mainwindow.h: fichero de código creado automáticamente a partir del mainwindow.ui.

Makefile, Makefile.Debug, Makefile.Release: ficheros makefile del proyecto, en modo *debug* o *release*.

- Directorio debug: archivos de compilación del proyecto en modo depuración, código objeto (.o) y el ejecutable (.exe).
- Directorio release: lo mismo pero para el modo versión final.

UN SENCILLO CONTADOR

Ahora realizar algo un poco más avanzado: un contador manual. Crear un proyecto nuevo (repetir los pasos del 7 al 14) de tipo **QWidget** y le da el nombre **Contador**. (¡Cuidado! No meterlo como un subdirectorio dentro de HolaMundoQt. Si está abierto el proyecto HolaMundoQt, cerrarlo primero.)

En el modo **Edit**, ir al fichero **Forms | widget.ui**, y se abre el editor de formularios. Seleccionar componentes de la paleta de componentes y ponerlos en el formulario, hasta crear una ventana con el siguiente aspecto. Algunas propiedades deben ser ajustadas con el inspector de objetos.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

REVISION 1/1	
	PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS
CARRERA	INGENIERÍA ELECTRÓNICA
SEDE	GUAYAQUIL

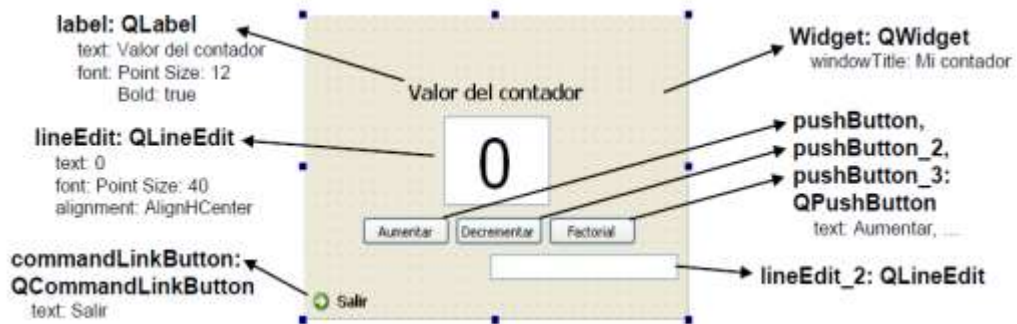


Figura 4.109 Editor de Formularios

Guardar y luego ejecutar para ver el aspecto que tendrá el programa.

Seguidamente meter el código asociado a los eventos de interés. Seleccionar el botón **Salir**, dar click en el botón derecho del ratón, y elegimos Go to slot... Luego seleccionar el evento **clicked()** y **OK**.

Ahora está dentro del editor de código, editando el fichero widget.cpp, y dentro del método creado **on_commandLinkButton_clicked()**. Escribir el siguiente código:

```
close();
```

La operación **close()** es un método de la clase **Widget**(dentro de la cual se encuentra), e indica que se cierre la ventana.

Ahora a programar el efecto de los botones. Para ello necesitamos una variable entera que sirve de contador. Para simplificar, la va a definir como una variable

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

global (es decir, fuera de la clase ventana). En widget.cpp va justo después de los *includes* y escribir:

```
int contador= 0;
```

Igual que se hace con el botón Salir, hacer con el botón Aumentar. Ir a widget.ui. Crear para este botón su evento clicked() y escribir:

```
void Widget::on_pushButton_clicked()
{
    contador++;
    ui->lineEdit->setText(QString::number(contador));
}
```

El atributo **ui** (*user interface*) está dentro de nuestro formulario (en la clase Widget) y a su vez **ui** contiene todos los elementos del formulario (se tiene ui->label, ui->lineEdit, ui->pushButton, etc.). Por otro lado, **QString::number** es una operación estática para convertir un número en un QString (el tipo cadena usado en Qt).

De la misma forma, para el botón pushButton_2 le asocia el evento:

```
void Widget::on_pushButton_2_clicked()
{
    contador--;
    ui->lineEdit->setText(QString::number(contador));
}
```

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Finalmente, quiere que el botón pushButton_3 calcule el factorial del valor actual del contador. Así que en el código de su evento escribir:

```
void Widget::on_pushButton_3_clicked()
{
    long resultado= 1;
    for (int i= 1; i<=contador; i++)
        resultado*= i;
    ui->lineEdit_2->setText(QString::number(resultado));
}
```

Guardar el proyecto y ejecutar. (Por cierto, ¿qué ocurre si ejecuta sin guardar primero?) Si hay problemas, repasar todos los pasos y preguntar al profesor.

RECURSOS UTILIZADOS

- Computadores Embebidos Raspberry-Pi y ODROID-U3
- Qt-Creator
- Guía de la practica

5. CONCLUSIONES.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

		REVISION 1/1
		PROCEDIMIENTO PARA EL DESARROLLO DE PRÁCTICA No 8
LABORATORIO	ELECTRÓNICA-EMBEBIDOS	
CARRERA	INGENIERÍA ELECTRÓNICA	
SEDE	GUAYAQUIL	

Al terminar esta práctica el alumno de Ingeniería Electrónica debe estar en la capacidad de establecer sus propias conclusiones.

Elaborado por: Daniel Martillo /Erly Zambrano	Revisado por: Ing. Luis Córdova	Aprobado por: Ing. Víctor Huilcapi Subía
Fecha de Elaboración 08-04-2015	Fecha de Revisión	Número de Resolución Consejo de Carrera:

CONCLUSIONES

La culminación de este trabajo de titulación deja la grata satisfacción de haber realizado una investigación profunda sobre temas innovadores de nuevas tecnologías presentes en la actualidad.

Los PLC's llegaron a la industria para proporcionar una herramienta de control industrial apropiada para enfrentar los desafíos tecnológicos del control industrial moderno y por ello se han ganado su espacio en la industria que es su principal ventaja.

Sin embargo hoy en día y con el significativo aumento de la calidad de la tecnología, tanto en la electrónica como del software, ese reinado de los PLC's en la industria se ha ido disminuyendo en el tiempo, debido a que existen sistemas embebidos tan o más eficientes y con precios muy atractivos.

Sin duda, que los sistemas embebidos en el ámbito del control industrial abarcan cada vez más terreno y paulatinamente aumentan su penetración en el mercado en forma sostenida, sobre todo si la técnica sigue en proporción de mayores y mejores prestaciones.

Además de su uso personal y como plataforma para juegos, Raspberry-pi y ODROID-U3 son ideales como dispositivos de control industrial, señalización digital, investigación de interfaces, debido a su alto potencial, bajo coste y plataforma abierta que permite extensas modificaciones.

Raspberry Pi y ODROID-U3 es en la actualidad el mejor hardware compatible, y el más práctico para los proyectos y la educación electrónica, sin duda el más asequible... y se supone que eso es lo que importa.

En el desarrollo de este trabajo se logra descubrir que el software libre es una herramienta para la creatividad y desarrollo de la tecnología ya que cualquiera dispone de la capacidad de modificarlo y mejorarlo.

Una de las razones más importantes por la cual el uso del software libre no es tan popular, es la costumbre que se ha tenido desde la revolución de las tecnologías de información de utilizar software privativo.

Otro factor importante que no contribuye al uso del software libre, es la ignorancia de las personas y la falta de familiaridad con este tipo de software, por lo que a la hora de utilizarlo se torna más difícil su empleo en los usuarios.

El futuro es muy prometedor a medida que la denominada “Informática Exascale” marca el inicio de un nuevo y emocionante contexto en las tecnologías SoC embebidas.

RECOMENDACIONES

Como recomendaciones principales para los estudiantes que hagan uso de los sistemas embebidos Raspberry-pi y Odroid- U3 es que se trate de elaborar prácticas más complejas que estimulen el desarrollo de aplicaciones orientadas a cubrir nuevas necesidades que se presenten en el campo de la ingeniería electrónica.

Se recomienda también que los estudiantes se animen a utilizar en mayor grado el software libre, que no tengan miedo de investigar y probarlos, para que se den cuenta de los grandes usos que se les puede dar.

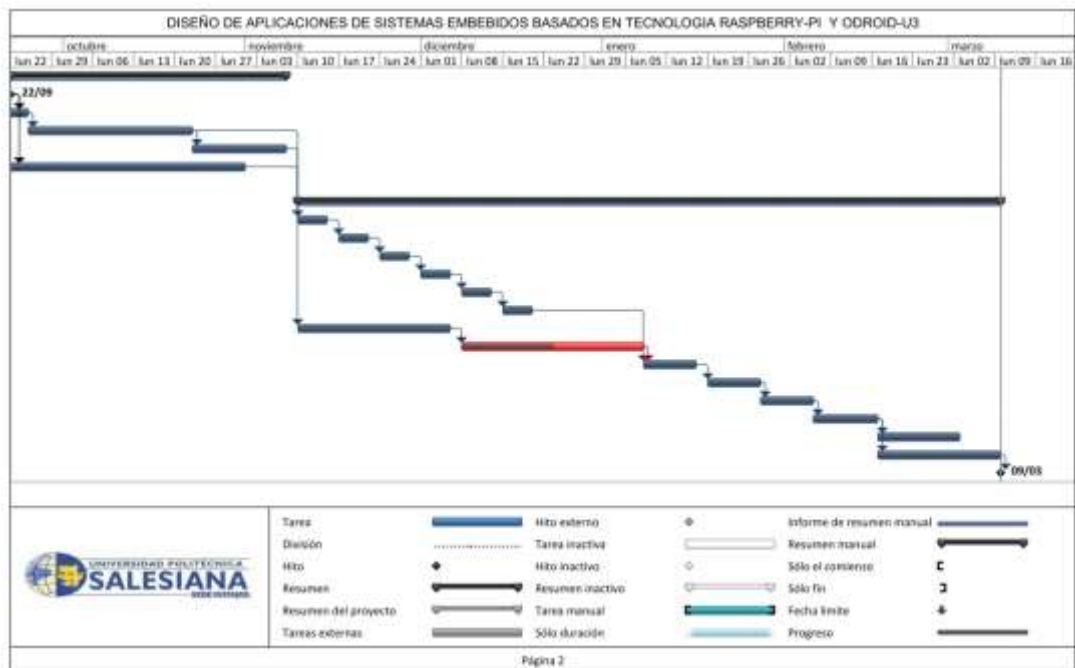
No existen limitaciones es solo de investigar y de tener una mente abierta para recibir nuevos conceptos y métodos en nuestros sistemas, lo cual les permitirá seguir avanzando en las nuevas tecnologías que hacen uso del software libre.

CRONOGRAMA

DISEÑO DE APLICACIONES DE SISTEMAS EMBEBIDOS BASADOS EN TECNOLOGIA RASPBERRY-PI Y ODROID-U3						
M	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	0	PRELIMINARES	35 días	lun 22/09/14	vie 07/11/14	
2	✓	Inscripción de trabajo de Tesis	0 días	lun 22/09/14	lun 22/09/14	
3	✓	Elaboración de listados de materiales y equipos para la tesis	3 días	lun 22/09/14	mié 24/09/14	1
4	✓	Pedido de importación de los embebidos ODROID (Alemania) Primer pedido.	20 días	jue 25/09/14	mié 22/10/14	3
5	✓	Compra Local de accesorios y Raspberry PI	12 días	jue 23/09/14	vie 07/11/14	4
6	✓	Consolidación de documentación, investigación y análisis de los embebidos Raspberry PI & ODROID	30 días	lun 22/09/14	vie 31/10/14	2
7	0	PRACTICAS	86 días	lun 10/11/14	lun 09/03/15	
8	✓	Ejecución y elaboración de practica 1	5 días	lun 10/11/14	vie 14/11/14	5,6
9	✓	Ejecución y elaboración de practica 2	5 días	lun 17/11/14	vie 21/11/14	8
10	✓	Ejecución y elaboración de practica 3	5 días	lun 24/11/14	vie 28/11/14	9
11	✓	Ejecución y elaboración de practica 4	5 días	lun 01/12/14	vie 05/12/14	10
12	✓	Ejecución y elaboración de practica 5	5 días	lun 08/12/14	vie 12/12/14	11
13	✓	Ejecución y elaboración de practica 6	5 días	lun 15/12/14	vie 19/12/14	12
14	✓	Pedido de importación de los embebidos ODROID (Alemania) Segundo pedido.	20 días	lun 10/11/14	vie 05/12/14	4
15	✓	Pedido de importación de los embebidos ODROID (Alemania) Tercer pedido.	23 días	lun 08/12/14	mié 07/01/15	14
16	✓	Ejecución y elaboración de practica 7	7 días	jue 08/01/15	vie 16/01/15	15,13
17	✓	Ejecución y elaboración de practica 8	7 días	lun 19/01/15	mar 27/01/15	16
18	✓	Ejecución y elaboración de practica 9	7 días	mié 28/01/15	jue 05/02/15	17
19	✓	Ejecución y elaboración de practica 10	7 días	vie 06/02/15	lun 16/02/15	18
20	✓	Compra de maletas y accesorios para presentación	10 días	mar 17/02/15	lun 02/03/15	19
21	✓	Elaboración de documento final para primera presentación al consejo.	15 días	mar 17/02/15	lun 09/03/15	19
22	✓	Presentación del primer borrador.	0 días	lun 09/03/15	lun 09/03/15	21

	Tarea: Hito externo División: Tarea inactiva Hito: Hito inactivo Resumen: Resumen inactivo Resumen del proyecto: Tarea manual Tareas externas: Sólo duración	Informe de resumen manual: Resumen manual: Sólo el comienzo: Sólo fin: Fecha límite: Progreso:
---	---	---

Página 1



PRESUPUESTO

TEMA: DISEÑO DE APLICACIONES DE SISTEMAS EMBEBIDOS BASADOS
EN TECNOLOGÍA RASPBERRY-PI Y ODROID-U3

Nombre: Erly Zambrano/Daniel Martillo

Tutor: Ing. Luis Córdova

Fecha: 23/04/2015

ITEM	DESCRIPCION	QTY	VALOR U	TOTAL
EMBEBIDOS ODROID				
1	ODROID U3	10	\$ 103,94	\$ 1.039,35
2	ODROID U3 CASE	10	\$ 6,44	\$ 64,35
3	ODROID U3 IO SHIELD	10	\$ 29,84	\$ 298,35
4	Plug in Unidad de potencia 5v 2A	10	\$ 10,00	\$ 100,00
5	ODROID U3 Cooling Fan U3	10	\$ 8,39	\$ 83,85
6	MICRO SD 16 Gb con adaptador	10	\$ 12,00	\$ 120,00
7	Cable MICRO HDMI	10	\$ 9,00	\$ 90,00
8	Convert HDMI a VGA+	5	\$ 25,00	\$ 125,00
9	Web CAM USB (Genérica)	10	\$ 12,00	\$ 120,00
10	Modem Wi-Fi TP-LINK TL-WN722N	10	\$ 30,00	\$ 300,00
Subtotal de ODROID				\$ 2.340,90
EMBEBIDOS RASPBERRY PI				
11	Raspberry PI	10	\$ 70,00	\$ 700,00
12	Caja de Raspberry PI CASE	10	\$ 8,00	\$ 80,00
13	Cámara para Raspberry PI	10	\$ 40,00	\$ 400,00
14	MICRO SD 16 Gb con adaptador	10	\$ 12,00	\$ 120,00
15	Cable HDMI 1.5,	10	\$ 8,00	\$ 80,00
16	Convert HDMI a VGA+	5	\$ 25,00	\$ 125,00
17	Modem Wi-Fi TP-LINK TL-WN725N	10	\$ 25,00	\$ 250,00
Subtotal de RASPBERRY PI				\$ 1.755,00
ACCESORIOS COMUNES				
18	Mouse pequeño genérico	10	\$ 8,00	\$ 80,00
19	Teclado 105 TECLAS genérico pequeño	10	\$ 13,00	\$ 130,00
20	Caja metálica Presentación	10	\$ 59,00	\$ 590,00
21	Cable USB a pin 1,5 mm (Alimentación ODROID)	10	\$ 5,00	\$ 50,00
22	Cable USB a micro USB (Alimentación Raspberry)	10	\$ 5,00	\$ 50,00
ACCESORIOS COMUNES				\$ 900,00
GASTOS DE IMPORTACION				
23	Gastos de importación # 1	1	\$ 69,00	\$ 69,00
24	Gastos de importación # 2	1	\$ 198,00	\$ 198,00
25	Gastos de importación # 3	1	\$ 69,00	\$ 69,00
Gastos de Importación				\$ 336,00
SUBTOTAL				\$ 5.331,90
IVA				\$ 639,83
TOTAL				\$ 5.971,73

BIBLIOGRAFÍA

Raspberry-Pi guía del usuario por Eben Upton (cocreador de la Raspberry) y Gareth Halfacree.

The Raspberry Pi Education Manual

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.

Version 1.0 December 2012

the MagPi , The MagPi is the official Raspberry Pi magazine, números del 1 al 30, <https://www.raspberrypi.org/magpi/>

Revista ODROID Magazine. <http://magazine.odroid.com/>

Comandos Linux unix y programación shell

Documento de la charla de Iñigo Tejedor & Pello Altadill

<http://4party.cuatrovientos.org>

Python para todos

Por Raúl González Duque

Este libro se distribuye bajo una licencia Creative Commons Reconocimiento 2.5 España.

Snake Wrangling for Kids, Learning to Program with Python

Por Jason R. Briggs

Versión 0.7.7

Copyright c 2007.

Qt Reference Documentation (Open Source Edition),

<http://doc.trolltech.com/4.3/>

Web oficial del OpenGL consortium

<http://www.opengl.org>

J. Blanchette y M. Summerfield, **C++ GUI Programming with Qt 4**, 2nd Edition, Prentice Hall, 2008 (Versión online desde la biblioteca en Safari: <http://safari.oreilly.com/9780137143979>).

An introduction to design patterns in C++ with Qt4--- Prentice Hall ---Alan & Paul Ezust.

C++ GUI Programming with Qt4 --- Prentice Hall --- Jasmin Blanchette & Mark Summerfeld.

Procesamiento de Imágenes Guión de Prácticas Sesión 1. Programación visual con Qt Creator ; Universidad de Murcia , Facultad de Informática Campus de Espinardo – 30100.

Raspberry Pi: Programando el puerto GPIO con Qt Creator .
Biblioman www.aquihayapuntes.com 18/09/2013.

Matemáticas en Ingeniería con Matlab y Octave ,Release 0.1 ,Guillem Borrell i Nogueras et al.
12 de October de 2013

Introducción a Octave Documentation
Release 1.0, Santiago Higuera, **January 17, 2015**

Scilab. Programación y Simulación, Calvo Rolle, José Luis, editorial RA-MA, EDITORIAL 2009.

Scilab (a Free Software to Matlab), S Chang & Co Ltd, 2001.

<http://www.abc.es/tecnologia/informatica-hardware/20130716/abci-raspberry-como-201307151936.html>.

ANEXOS

Raspberry Pi , características técnicas

	RPI Model A	RPI Model A+	RPI Model B	RPI Model B+	RPI 2 Model B
SoC	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836
CPU	ARM11 ARMv6 700 MHz.	ARM11 ARMv6 700 MHz.	ARM11 ARMv6 700 MHz.	ARM11 ARMv6 700 MHz.	ARM11 ARMv7 ARM Cortex-A7 4 núcleos @ 900 MHz.
GPU	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0
RAM	256 MB LPDDR SDRAM 400 MHz.	256 MB LPDDR SDRAM 400 MHz.	512 MB LPDDR SDRAM 400 MHz.	512 MB LPDDR SDRAM 400 MHz.	1 GB LPDDR2 SDRAM 450 MHz.
USB 2.0	1	1	2	4	4
Salidas de vídeo	HDMI 1.4 @ 1920x1200 píxeles	HDMI 1.4 @ 1920x1200 píxeles	HDMI 1.4 @ 1920x1200 píxeles	HDMI 1.4 @ 1920x1200 píxeles	HDMI 1.4 @ 1920x1200 píxeles
Almacenamiento	SD/MMC	microSD	SD/MMC	microSD	microSD
Ethernet	No	No	Sí, 10/100 Mbps	Sí, 10/100 Mbps	Sí, 10/100 Mbps
Tamaño	85,60x56,5 mm	65x56,5 mm.	85,60x56,5 mm	85,60x56,5 mm	85,60x56,5 mm
Peso	45 g.	23 g.	45 g.	45 g.	45 g.
Precio	25 dólares	20 dólares	35 dólares	35 dólares	35 dólares

GLOSARIO

ARM

La arquitectura ARM describe un tipo de procesadores diseñados y licenciados por la compañía británica ARM Holdings. Esta compañía no fabrica los chips, si no que vende los planos de cómo fabricarlos a cualquier empresa y está ya se encarga de hacérselos ella misma o un tercero. Este tipo de procesadores los usa apple, Samsung o Broadcom.

ARM11

ARM11 es lo que se denomina familia dentro de la arquitectura ARM. Es de 32-bits, en concreto la versión 6 o ARMv6. Esta arquitectura solo incluyó esta familia. En 2013 ya van por la arquitectura ARMv8 con las familias Cortex-A53 y Cortex-A57. Todos los móviles actuales usan familias superiores al menos basadas en ARMv7 como los iPhone.

ARMv6

ARMv6 es la versión número 6 de la arquitectura ARM de microprocesadores y es la que incluyen todos los de la familia denominada ARM11.

ARMv7

Esta arquitectura de procesadores de la familia ARM cuenta con varias ramas, siendo la más amplia e importante la ARMv7-A de 32 bits y que incluye los siguientes diseños de núcleos o cores: ARM Cortex-A5, ARM Cortex-A7, ARM Cortex-A8, ARM Cortex-A9, ARM Cortex-A12, ARM Cortex-A15 y ARM Cortex-A17.

Arquitectura Harvard

Originalmente, el término Arquitectura Harvard hacía referencia a las arquitecturas de computadoras que utilizaban dispositivos de almacenamiento físicamente separados para las instrucciones y para los datos (en oposición a la Arquitectura de von Neumann).

Actualmente este tipo de arquitectura se utiliza en los procesadores de tipo DSP como el VideoCore IV de la Raspberry Pi.

Arquitectura von Neumann

La arquitectura de von Neumann es una familia de arquitecturas de computadoras que utilizan el mismo dispositivo de almacenamiento tanto para las instrucciones como para los datos (a diferencia de la arquitectura Harvard). En general los microprocesadores de propósito general utilizan esta arquitectura, como los ARM o los de tipo x86.

BCM2835

BCM2835 es el microprocesador de tipo system on a chip (SoC) que lleva la Raspberry Pi y que incluye un núcleo ARM1176JZF-S a 700 MHz que puede llegar 1 GHz haciendo overclock y una GPU VideoCore IV.

BCM2836

BCM2836 es el microprocesador de tipo system on a chip (SoC) que lleva la Raspberry Pi 2 y que incluye cuatro núcleos Cortex-A7 a 800 MHz que puede llegar 900MHz haciendo overclock y una GPU VideoCore IV.

big.LITTLE

big.LITTLE es una arquitectura de computadores heterogénea desarrollada por ARM. Esta arquitectura se compone de dos tipos de procesadores unos denominados big, más potentes y que consumen más energía y otros denominados LITTLE menos potentes pero que consumen menos.

Broadcom

Broadcom Corporation es uno de los principales fabricantes de circuitos integrados para comunicaciones de banda ancha de los Estados Unidos. No cuenta con fábricas, si no que desarrollo los circuitos y los encarga a otras empresas. Algunos de sus diseños están basados en los procesadores de la arquitectura ARM, como el BCM2835 que incluye la Raspberry Pi.

Raspberry Pi.

Cortex-A5

Cortex-A5 es el núcleo concreto dentro de lo que se denomina familia, en particular ARM Cortex-A como una de las versiones disponibles de la arquitectura ARM. Este núcleo puede montarse en configuraciones single, dual core o quad core.

Cortex-A7

Cortex-A7 es el núcleo concreto dentro de lo que se denomina familia, en particular ARM Cortex-A como una de las versiones disponibles de la arquitectura ARM. Este núcleo puede montarse en configuraciones single, dual core o quad core.

CPU

La CPU (Central Processing Unit) o Unidad Central de Procesamiento, o simplemente el procesador o microprocesador, es el componente principal del ordenador y otros dispositivos programables, que interpreta las instrucciones contenidas en los programas y procesa los datos. Cada tipo de CPU tiene un tipo diferente de instrucciones y por lo tanto los programas que vaya a usar se tienen que preparar específicamente para esa CPU. Uno de estos programas, el más importante, es el Sistema Operativo que para poder funcionar en la Raspberry Pi tiene que estar preparado, o sea, programado y compilado para ejecutarse en un procesador de tipo ARM de la familia ARM11 con el core ARMv6. Esta arquitectura es bastante obsoleta pero suficiente, sin embargo es uno de los lastres de la Raspberry Pi ya que la mayoría de los programas y versiones que habías hasta ahora eran para ARMv7 también conocidos como Cortex e incluidos en muchos teléfonos móviles.

Debian

Debian es un sistema operativo (S.O.) libre.

El sistema operativo es el conjunto de programas básicos y utilidades que hacen que funcione tu ordenador o computadora. Una de las versiones disponibles contiene el núcleo de Linux. Hay disponible una versión especial para las Raspberry Pi denominada Raspbian

DSP

DSP o Digital Signal Processor es un procesador digital de señales basado en un procesador o microprocesador que posee un conjunto de instrucciones, un hardware y un software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad. Orientados especialmente al procesamiento digital de muchos datos, como el audio o el vídeo, pero con pocas instrucciones muy repetitivas sobre ellos, por lo que utilizan la denominada Arquitectura Harvard en vez de la von Neumann de los procesadores de propósito general.

eMMC

eMMC son las siglas en inglés de embedded Multi-media Card. O sea, tarjeta de memoria multimedia empotrada. Se trata de la combinación del chip de memoria igual a una tarjeta SD o MMC, pero incluyendo el controlador. Esto, más o menos, hace que al estar todo integrado la velocidad de esta memoria sea mejor que la de una tarjeta SD por separado.

Este tipo de memoria se encuentra incluido en muchas placas, como la nueva Raspberry pi compute module.

Entorno gráfico

El Entorno Gráfico, interfaz gráfica de usuario o GUI (del inglés Graphical User Interface) es un programa informático que actúa de intermediario entre el ordenador y el usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.

fat32

FAT, del inglés file allocation table) o tabla de asignación de archivos es un sistema de archivos desarrollado para MS-DOS y que continúa usándose con los sistemas Microsoft Windows.

FAT32 fue la solución para superar el límite de tamaño del sistema original denominado FAT o FAT16 y manteniendo compatibilidad con MS-DOS. El tamaño máximo de un archivo en FAT32 es 4GB (2³²-1 bytes).

Firmware

Firmware es un programa informático para usos concretos. Se encuentra instalado normalmente una memoria interna de lectura/escritura (ROM, EEPROM, flash, etc.), *y que se conecta a bajo nivel con el hardware.*

FPGA

Una FPGA (del inglés Field Programmable Gate Array) o Matriz de Puertas Programable en Campo o in situ, es un dispositivo electrónico que contiene una matriz de bloques lógicos o puertas lógicas cuya interconexión y funcionalidad puede ser configurada “in situ” mediante un lenguaje de descripción especializado, o sea, puede ser “programada”.

Fundación Raspberry Pi

La Fundación Raspberry Pi sin ánimo de lucro se fundó en 2009 para promover los estudios de informática y programación básica en las escuelas. Es responsable del desarrollo del computador todo en uno llamado Raspberry Pi.

GPIO

General Purpose Input/Output (GPIO), en nuestro caso particular son los pines laterales de la Raspberry Pi, hay que distinguir ciertos cambios entre la versión inicial y la actualizada, rev 1 y rev 2, respectivamente.

GPU

La GPU (acrónimo del inglés Graphics Processing Unit) o unidad de procesamiento gráfico es un coprocesador dedicado al procesamiento de gráficos u operaciones de coma flotante, para aligerar la carga de trabajo del procesador central o CPU en aplicaciones como los videojuegos y o aplicaciones 3D interactivas y permitir acelerar estos procesos.

Muchas veces se habla de la GPU como la aceleradora 3D o acelerador 3D.

H.264

H.264 o MPEG-4 parte 10 es una norma que define un códec de vídeo de alta compresión, desarrollada conjuntamente por el ITU-T Video Coding Experts Group

HDMI

HDMI es el acrónimo de High-Definition Multimedia Interface, es un conector de tipo Digital, para audio y vídeo y un protocolo para su envío y recepción.

VCEG) y el ISO/IEC Moving Picture Experts Group (MPEG).

Linux

GNU/Linux o solo Linux como se denomina para abreviar, es un sistema operativo basado en software libre muchas de cuyas herramientas forman parte del proyecto GNU.

Lo que se denomina núcleo o kernel es la parte central del sistema operativo y fue creado originalmente por Linus Torvalds como un proyecto personal que se puso a disposición de todo el mundo a través de Internet.

Linux se puede usar con entorno gráfico de tipo ventanas o sin él, o sea, en modo consola o terminal.

Linux se encuentra agrupado con otros programas útiles en lo que se denominan distribuciones, ejemplos de distribuciones son:

Debian

Raspbian

Arch Linux

Fedora

SuSE

Ubuntu

LXDE

LXDE es un entorno gráfico libre para Unix y otras plataformas POSIX, como Linux o BSD. El nombre corresponde a Lightweight X11 Desktop Environment, que en español significa Entorno de escritorio X11 ligero.

LXDE es un proyecto que trata de hacer disponible un nuevo entorno de escritorio ligero y rápido. No está diseñado para ser tan complejo como KDE o GNOME, pero es bastante usable y ligero, y mantiene una baja utilización de recursos.

NFC

Son las siglas en inglés de Near Field Communication, comunicaciones de cercanas de campo y se refiere a la tecnología para leer y escribir datos de forma inalámbrica y sin contacto. Las tecnologías utilizadas se denominan comúnmente RFID.

Concretamente, NFC se refiere a corto alcance de unos 10 cm y usa las frecuencias de 13.56 MHz. Sus características están estandarizadas por el NFC Forum y están incluidas en la norma internacional ISO/IEC 14443.

NTFS

NTFS, del inglés New Technology File System, es un sistema de archivos de Windows NT incluido en las versiones profesionales y domésticas a partir de XP.

El tamaño máximo de archivo que permite es de ****16TB****.

OpenGL

OpenGL (Open Graphics Library) es una especificación estándar que define un sistema para programar aplicaciones que produzcan gráficos 2D y 3D.

OpenGL ES

OpenGL ES (OpenGL for Embedded Systems) es una variante simplificada del sistema para programar gráficos 3D de tipo OpenGL diseñada para dispositivos integrados tales como teléfonos móviles, PDAs y consolas de videojuegos o la Raspberry Pi.

OpenVG

OpenVG (Open Vector Graphics) es una especificación estándar que define un sistema para programar aplicaciones que produzcan gráficos 2D acelerados y que se complementa con OpenGL ES para los gráficos 3D.

Overclock

Overclock u *overclocking* es una forma de configurar un procesador o sistema para usar una mayor velocidad de reloj. De esta forma, al ir más rápido el reloj que sirve de referencia para marcar el ritmo al que se ejecutan las instrucciones y el sistema operativo en conjunto, mejora el rendimiento.

PCB

PCB son las siglas en inglés de Printed Circuit Board, o sea, “circuito impreso” o “tarjeta de circuitos impresos” o “placa de circuitos impresos”.

Premier Farnell

Premier Farnell, PLC. es un distribuidor británico de componentes electrónicos como por ejemplo la Raspberry Pi.

También opera con los nombres:

Farnell en Reino Unido y resto de Europa

element14 en Australia, Nueva Zelanda, Hong Kong, India, Malasia, China y Singapur

Newark Electronics en the US, Canada and Mexico

RFID

RFID son las siglas en inglés de Radio Frequency IDentification, y es un conjunto genérico de tecnologías para lectura o intercambio de datos de forma inalámbrica sin contacto. El rango de operación va desde pocos centímetros hasta varios metros, las tecnologías implicadas se denominan como: LF (120 to 150 KHz), HF (13.56 MHz), UHF (433 to 900 MHz), y están estandarizadas en la norma internacional ISO 18000.

RS Components

RS Components, también conocida como RS, es un distribuidor mundial de componentes electrónicos como por ejemplo la Raspberry Pi, con sede en Corby en el Reino Unido.

RTC

RTC significa real time clock y se refiere a los dispositivos que tienen un reloj interno con batería que guarda la hora cuando está apagado. La Raspberry Pi no posee RTC por lo que coge la hora a través de Internet cada vez que se enciende y se conecta. Si no se conecta, se muestra la hora por defecto, 30 de Noviembre de 1999 a las 19:30 o algo así, y a partir de ahí continuará. Hay proyectos para añadir un reloj (RTC) a la Raspberry Pi.

SBC

Un Single-Board Computer es un ordenador o computador completo fabricado en una sola placa de circuito impreso o PCB. Esta placa debe incluir: el procesador normalmente de tipo SoC, la memoria RAM, los sistemas de entrada y salida mediante periféricos, o incluso la memoria permanente de tipo flash RAM que funciona como disco duro.

SFTP

SSH File Transfer Protocol (también conocido como SFTP o *Secure File Transfer Protocol*) es un protocolo del nivel de aplicación que proporciona la funcionalidad necesaria para la transferencia y manipulación de archivos de forma segura.

Sistema Operativo

Un sistema operativo (SO, frecuentemente OS, del inglés Operating System) es un programa o conjunto de programas con que un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación, ejecutándose en modo privilegiado respecto de los restantes. Ejemplos de Sistemas Operativos son:

Linux

O GNU/Linux el principal sistema disponible para la Raspberry Pi

SoC

System on Chip o system on a chip es un ordenador completo en un solo chip o circuito integrado. Suele incluir el procesador, la tarjeta de vídeo, la de sonido, etc., el resto de la placa sobre la que se monta suele incluir las conexiones y los sistemas adicionales que no se hayan incluido en el chip como por ejemplo: tarjeta de red, wifi, bluetooth, etc.

La Raspberry Pi incluye un chip de tipo SoC con arquitectura ARM fabricado por Broadcom, en concreto el modelo BCM2835 con procesador principal o CPU ARM1176JZF-S y procesador gráfico o GPU VideoCore IV.

SoM

SoM son las siglas en inglés de System on Module es un ordenador y placa base completa en un solo módulo. Suele incluir el procesador, la tarjeta de vídeo, la de sonido, etc., como en un SoC pero añadiendo la memoria RAM y el disco duro en formato Flash o eMMC.

La nueva Raspberry Pi compute module es un sistema de este tipo.

SSH

SSH (Secure SHell, en español: intérprete de órdenes segura) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el entorno gráfico para poder ejecutar programas gráficos mediante lo que se llama exportar las X. SSH también trae incluido un servicio para el intercambio de ficheros denominado sFTP.

Tarjeta SD

Secure Digital (SD) es un formato de tarjeta de memoria inventado por Panasonic. La Raspberry Pi utiliza este tipo de tarjeta para almacenar el sistema de arranque (la típica BIOS de los PC) y el sistema operativo.

Terminal

La Terminal o Interfaz de Línea de Comandos (CLI), por su acrónimo en inglés de Command Line Interface, es una forma de introducir instrucciones en algún programa informático o sistema operativo por medio de una línea de texto simple.

Esta forma de interactuar con un ordenador requiere menos recursos que una interfaz gráfica pero es menos visual y más compleja de utilizar, ya que obliga a teclear los comandos que queremos ejecutar, y debemos saber la sintaxis exacta de estos y sus correspondientes opciones.

VideoCore IV

VideoCore es un procesador multimedia de bajo consumo desarrollado originalmente por Alphasoic Ltd que es ahora propiedad de Broadcom. Es un procesador de tipo

DSP de dos dimensiones con capacidad para codificar y decodificar archivos multimedia manteniendo el bajo consumo.

La versión I de VideoCore se incluía en varios teléfonos móviles Samsung, la versión II se incluye en los iPod de 5ª generación y el III se utiliza en el Nokia N8. El VideoCore IV es el procesador gráfico de la Raspberry Pi soportando codificación y decodificación de vídeos a 1080p, es el mismo que incluye el Nokia 808 PureView.

VPU

La VPU (acrónimo del inglés Video Processing Unit) o unidad de procesamiento de vídeo es en realidad una habilidad de la GPU para decodificar vídeo sin hacer uso del procesador central o CPU

Muchas veces se habla de las capacidades de la GPU en modo VPU como la “aceleradora de vídeo” o “acelerador de vídeo” o en inglés como “GPU accelerated video decoding”, “GPU assisted video decoding”, “GPU hardware accelerated video decoding” o “GPU hardware assisted video decoding”.

XBMC

XBMC (abreviatura de “Xbox Media Center”) es un centro multimedia o “media center” de entretenimiento multiplataforma bajo la licencia GNU/GPL. Su nombre se debe a que inicialmente fue creado para la primera generación de la videoconsola Xbox. Posteriormente, el equipo de desarrollo de XBMC ha portado el producto para que pueda correr de manera nativa en Linux (incluyendo las versiones específicas para Raspberry Pi: Raspbmc y Xbian), Mac OS X y Windows.