Carrera:	BIOINGENIERÍA	N° de orden:	10
Asignatura:	Programación II	Horas cátedras semanales:	5
Departamento	Bioingeniería	Horas reloj total	120
Bloque	Tecnologías Básicas	Nivel	2
Área	Digital		
Competencias	**Genéricas**	**Específicas**	
	CE 1 Diseñar, Calcular y Proyectar instalaciones, equipamiento e instrumental biomédico, aplicando conocimiento integral y tecnologías adecuadas para atender la demanda de la población y las variables económicas características de la bioingeniería. Nivel 1 **CE 2** Proyectar, dirigir y controlar la construcción, operación y mantenimiento de instalaciones, equipamientos e instrumental de tecnología biomédica, procesamiento de señales biomédicas y sistemas derivados de biomateriales utilizados en el área de la salud. Nivel 1 **CT5:** Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas. Nivel 1 **CS5:** Actuar con espíritu emprendedor. Nivel 1		

Objetivos

Que los y las estudiantes sean capaces de:

- Diseñar y desarrollar aplicaciones para resolución de problemas complejos aplicados a ingeniería electrónica.
- Utilizar fluidamente herramientas de documentación, de control de versiones, y de automatización de la construcción de un programa o biblioteca a partir de las fuentes.

Contenidos que se trabajan en la actividad (Mínimo)

Unidad/Módulo	Carga Horaria (h)
UNIDAD 1: El Lenguaje C++ y las Estructuras Avanzadas de Datos Introducción a la programación orientada a Objetos - Lenguaje C++ como evolución respecto del lenguaje C - Tipos abstractos de datos: Clases - Clases y Objetos - Miembros públicos y privados - constructores y destructores - Concepto de encapsulamiento. Entrada/Salida en C++.	25

Operador visibilidad - Operadores para gestión dinámica de memoria: new y delete - Sobrecarga de funciones y operadores. Funciones y clases friend - uso del apuntador this - calificador const - especificador de clase de almacenamiento: extern y static. Herencia: tipos (hincapié en herencia pública) - clase base y derivada – miembro protected – constructores y destructores en la clase derivada. Contenedores de datos – Aplicaciones con Estructuras Avanzadas de Datos: Colas, Pilas, Listas. Funciones Virtuales y Virtuales puras – Clases Abstractas – Concepto de Interface	
UNIDAD 2: Entornos Gráficos Introducción a la programación en entornos gráficos - Bibliotecas y entornos de desarrollo: Caso particular de QT y el QT Creator. -Introducción a la programación gobernada por eventos y las señales (Signals and Slots) de QT. - Formularios Básicos: Cuadros de Diálogo y aplicaciones de formularios sencillas. - Widgets más comunes: Button, Check box, Radio button, Menu bar, Toolbar, Scrollbar, Text box, Combo box, Label y Otros. Ejemplos de aplicación: Clientes de sistemas computacionales (PC , bases de datos, Sistemas embebidos, etc.) comunicación serie, representaciones gráficas (data logger)	12
UNIDAD 3: Introducción a los microcontroladores y los Sistemas Embebidos. Concepto de Sistema Embebido. Presentación de periféricos elementales: Puertos de entrada/salida, Contador, Timer, UART, etc. IDE para microcontroladores - Compilación modular - Metodología de proyectos – Diagrama de capas: drivers, primitivas, aplicación – portabilidad. Concepto de Sistema Operativo en Tiempo Real. DIAGRAMA DE CAPAS: Metodología de trabajo. Aplicación - Primitivas – Drivers.	6
UNIDAD 4: Introducción a la arquitectura de un Microcontrolador. Caso de uso: **Cortex M0+ y su programación en C++** Microcontrolador: Análisis de un circuito inteligente básico. Integración de periféricos en un solo chip. Familia Córtex M0+. Presentación del mapa de memoria y periféricos. Técnicas de acceso. Identificación de Registros del Core y de sus periféricos. Especificadores de memoria. La Switch Matrix (SWM), registros y operación.	6
UNIDAD 5: Introducción a GPIO.	6

GPIO (General Purpose Input Output): Configuración de los puertos de entrada/salida de propósito general: tratamiento como entrada, tratamiento como salida, tratamiento bidireccional. Registros asociados. Lectura de entradas digitales simples - Acción sobre salidas digitales. Consideraciones teóricas de las entradas y las salidas de los microcontroladores. Desarrollo de funciones utilitarias en C++ para el manejo de GPIO.	
UNIDAD 6: Pooling (encuesta) e Interrupciones Idea general - Estrategias de atención - Funciones de interrupción - Pooling vs. Interrupciones - Registros asociados - Vector de interrupciones - Prioridades - Distribución de tiempos de ejecución para el programa principal y las funciones de interrupción - Mínimo tiempo admisible entre interrupciones - Pasaje de información entre el programa principal y las funciones de interrupción. Fuentes interrupción externas (en pines).	12
UNIDAD 7: Contadores / Temporizadores Contador - Características principales – Temporizador: Caso particular de contador - Utilización como Contador (acumuladores de eventos) - Utilización como Temporizador (Base de Tiempo) - Multiplicadores de la base de Tiempo (ticks). Implementación - Registros asociados - Modos de funcionamiento. scheduler: Principios de funcionamiento. Caso de estudio: El systick del core del Cortex M.	6
UNIDAD 8: GPIO avanzado. Desarrollo del firmware y primitivas asociadas a Display de 7 segmentos y a la lectura y depuración mediante técnicas de absorción de transitorios de teclados lineales, teclados matriciales y entradas digitales de microswitchs. Escritura de salidas digitales. Display LCD.	18
UNIDAD 9: Programación Gobernada por Eventos Introducción a los diagramas secuenciales con múltiples acciones temporizadas. Desarrollo de funciones facilitadoras: TimerStart(), TimerStop(), TimerClose(), etc. Máquina de Estados: Diagrama de globos. Implementación con switch-case. Implementación con múltiples if. Implementación con punteros a función. Modelización y Síntesis de problemas mediante la utilización de máquinas de estado - Casos de automatismos independientes	25

dentro de un mismo equipo - Utilización de máquinas de estados en paralelo: ventajas. Análisis de casos de aplicación típicos. Resolución de problemas mediante la combinación de Máquinas de estado y Diagramas secuenciales.	
UNIDAD 10: Comunicación Serie. Necesidad de la comunicación serie - Serializadores y paralelizadores – Comunicación serie asincrónica: Conceptos. Velocidad de transmisión - Registros asociados en el microcontrolador - Modos de operación – Buffers de Rx y Tx: Pilas y Colas circulares - Implementación de protocolos punto a punto y multipunto - Estrategias de programación por pooling e interrupciones. Desarrollo del firmware y primitivas asociadas a la comunicación serie.	25
UNIDAD 11: Introducción a la Medición y Generación de Señales Analógicas Introducción a la medición de magnitudes analógicas (temperatura, presión, humedad, etc.) - Conversores ADC: y DAC características principales - Registros asociados. Estrategias de programación por pooling y por interrupciones - Interpretación de los valores obtenidos (Tablas, filtros, etc.) - Eliminación de valores espurios: Filtros de media móvil y de mediana. Desarrollo del firmware y primitivas asociados a la medición y generación de señales analógicas.	13
UNIDAD 12: Métodos numéricos elementales Aplicaciones algorítmicas haciendo uso de métodos numéricos tradicionales. Soluciones aproximadas. Precisión. Errores. Ecuaciones diferenciales ordinarias: Ecuaciones con condiciones iniciales. Integración por Taylor. Método de Euler, Euler Gauss. Euler Richarson. Métodos de Runge-Kutte. Método predictor corrector. Estabilidad y convergencia. Sistemas de ecuaciones diferenciales. Ecuaciones de orden superior. Ecuaciones con condiciones de contorno. Métodos de diferencias finitas y elementos finitos para ecuaciones diferenciales.	12

Bibliografía

Bibliografía obligatoria, optativa y otros materiales del curso

[1] Red de desarrolladores de Qt en Español - MediaWiki, (06/03/19) https://wiki.qt.io/Main/es .
[2] M. J. Pont, (2002). Embedded C. Addison-Wesley.

[3] Anónimo. (2010). “APRENDA Qt4 DESDE HOY MISMO, https://www.academia.edu/21989349/APRENDA_Qt4_DESDE_HOY_MISMO

[4] B. W. Kernighan and D. Ritchie. (1991) El lenguaje de programación C. Prentice-Hall.

[5] J. M. GOMEZ, A. MARTIN PRAT, X. MOLINERO ALBAREDA, P. P. VAZQUEZ ALCOCER, and F. XHAFA.(2006) Programación en C++ para ingenieros. Madrid: Paraninfo SA..

[6] NXP® Semiconductors(Oct. 28, 2020) “Hoja de Datos LPC84x,”. https://www.nxp.com/docs/en/data-sheet/LPC84x.pdf

[7] H. M. DEITEL and P. J. DEITEL (2008), C/C++ Cómo programar, 6ta Edicion, 6th ed. Madrid: Pearson. www.deitel.com

[8] B. Hall (2020). “Beej’s Guide to Network Programming Using Internet Sockets,” p. 134, http://beej.us/guide/bgnet/pdf/bgnet_usl_c_1.pdf