



미래를 여는 기술 교육, CODEXPERT

2024년 개설 교육과정 상세소개

CONTENTS - 1/3

Part. 1

주요 교육 분야 및 과정 목록

- 06. C언어 Programming 분야
- 07. Programming 언어 분야
- 08. Data Science 분야
- 09. 인공지능 분야
- 10. Hardware 설계 분야
- 11. Embedded System 분야
- 12. Embedded Linux 분야

Part. 2

C언어 Programming 분야 교육 과정

- 14. C 프로그래밍
- 15. C 기반 문제 해결 기법
- 16. 게임을 만들며 배우는 임베디드 C
- 17. 실전 임베디드 C 프로그래밍
- 18. 고급 C 포인터 완전정복 및 데이터 모델링
- 19. 신뢰성과 최적화를 위한 고급 C 프로그래밍
- 20. C 기반 자료구조 프로그래밍
- 21. C 기반 Advanced 자료구조 프로그래밍
- 22. 알고리즘 기반 프로그램 설계 실무
- 23. 알고리즘 기반 프로그램 설계 전문가

Part. 3

Programming 언어 분야 교육 과정

- 25. C++ & OOP Programming
- 26. C++ STL Programming
- 27. C++ Design Pattern & Refactoring
- 28. Modern C++
- 29. Python Programming 완벽 이해
- 30. Python 기반 문제 해결 기법
- 31. 데이터 처리를 위한 Python 완벽 이해
- 32. Python 기반 알고리즘 프로그래밍
- 33. C# Programming
- 34. WPF를 이용한 GUI Programming
- 35. JAVA Programming
- 36. JAVA 기반 문제 해결 기법
- 37. 실무 Web Programming

CONTENTS - 2/3

Part. 4

Data Science 분야 교육 과정

- 39. 바로 실무에 써먹을 수 있는 SQL
- 40. Advanced SQL 활용 실무
- 41. SQL Tuning 실무
- 42. 데이터 분석을 위한 확률, 통계, 머신러닝
- 43. Xlwings를 이용 엑셀 업무 자동화
- 44. Pandas를 활용한 데이터처리
- 45. Python 모듈을 이용한 데이터 시각화
- 46. Python 모듈을 이용한 Dash Board 구현
- 47. PyQt를 이용한 GUI Programming

Part. 5

인공지능 분야 교육 과정

- 49. Python으로 배우는 통계학
- 50. 인공지능을 위한 선형 대수학
- 51. Scikit-learn을 활용한 머신러닝의 이해와 구현
- 52. Keras 기반 딥러닝의 이해와 구현
- 53. PyTorch 기반 딥러닝의 이해와 구현
- 54. Deep Learning 심화
- 55. 영상처리를 위한 CNN 실무
- 56. 시계열 처리를 위한 RNN 실무
- 57. 생성모델을 위한 GAN 실무
- 58. 강화학습 실무
- 59. 임베디드 리눅스 기반 온디바이스 AI
- 60. Non-OS 임베디드 시스템 기반 온디바이스 AI

Part. 6

Hardware 설계 분야 교육과정

- 62. SW 개발자를 위한 디지털 회로 기초
- 63. SW 개발자를 위한 임베디드 시스템 설계
- 64. Verilog와 FPGA를 이용한 하드웨어 설계 초급
- 65. Verilog와 FPGA를 이용한 다양한 Peripheral 설계

CONTENTS - 3/3

Part. 7

Embedded System 분야 교육 과정

- 67. 임베디드 시스템 필수 디바이스 프로그래밍
- 68. 임베디드 시스템 확장 디바이스 프로그래밍
- 69. 이더넷과 WiFi를 이용한 IoT디바이스 설계 실무
- 70. Arm Cortex-M 프로세서 시스템 프로그래밍
- 71. Arm Cortex-A 멀티코어 시스템 플랫폼 설계
- 72. FreeRTOS 커널 포팅 및 멀티태스킹 응용 프로그램 개발
- 73. OS 만들면서 배우는 Cortex-A 코프로세서 구조 및 제어

Part. 8

Embedded Linux 분야 교육 과정

- 75. 재미있게 배우는 임베디드 리눅스 활용
- 76. 실전 임베디드 리눅스 Application 프로그래밍
- 77. 리눅스 명령어 및 셸 프로그래밍
- 78. 리눅스 IPC의 이해와 응용
- 79. 임베디드 리눅스 커널 포팅 및 디바이스 드라이버
- 80. 임베디드 리눅스 멀티미디어 & 네트워크
- 81. 임베디드 시스템 및 리눅스 파일 시스템의 이해
- 82. 고급 임베디드 리눅스 디바이스 드라이버 설계

Part 1. 주요 교육 분야 및 과정 목록

* 본 목록은 대표 과목들로서 원하는 교육으로 변경 진행 가능하며 목록에 없는 교육도 과정 개발 또는 강사 소싱으로 진행 가능합니다

풍부한 개발 경험으로
실무 프로그래밍
노하우 전수

C 언어 Programming 분야

* 과목명을 누르면 과목 상세 소개 페이지로 이동합니다

언어1	수준	과목명	표준 시간	페이지
C	레벨1 / 초급	C 프로그래밍	40시간 / 5일	14
	레벨1 / 초급	C 기반 문제 해결 기법	40시간 / 5일	15
	레벨1 / 초급	게임을 만들며 배우는 임베디드 C	40시간 / 5일	16
	레벨2 / 중급	실전 임베디드 C 프로그래밍	40시간 / 5일	17
	레벨3 / 고급	고급 C 포인터 완전정복 및 데이터 모델링	40시간 / 5일	18
	레벨3 / 고급	신뢰성과 최적화를 위한 고급 C 프로그래밍	40시간 / 5일	19
	레벨2 / 중급	C 기반 자료구조 프로그래밍	40시간 / 5일	20
	레벨3 / 고급	C 기반 Advanced 자료구조 프로그래밍	40시간 / 5일	21
	레벨3 / 고급	알고리즘 기반 프로그램 설계 실무	40시간 / 5일	22
	레벨4 / 전문	알고리즘 기반 프로그램 설계 전문가	80시간 / 10일	23

풍부한 개발 경험으로
실무 프로그래밍
노하우 전수

Programming 언어 분야

* 과목명을 누르면 과목 상세 소개 페이지로 이동합니다

언어	수준	과목명	표준 시간	페이지
C++	레벨1 / 초급	C++ & OOP Programming	40시간 / 5일	25
	레벨2 / 중급	C++ STL Programming	40시간 / 5일	26
	레벨2 / 중급	C++ Design Pattern & Refactoring	40시간 / 5일	27
	레벨3 / 고급	Modern C++	40시간 / 5일	28
Python	레벨1 / 초급	Python Programming 완벽 이해	40시간 / 5일	29
	레벨1 / 초급	Python 기반 문제 해결 기법	40시간 / 5일	30
	레벨2 / 중급	데이터 처리를 위한 Python 완벽 이해	40시간 / 5일	31
	레벨2 / 중급	Python 기반 알고리즘 프로그래밍	40시간 / 5일	32
C#	레벨1 / 초급	C# Programming	40시간 / 5일	33
	레벨2 / 중급	WPF를 이용한 GUI Programming	40시간 / 5일	34
JAVA	레벨1 / 초급	JAVA Programming	40시간 / 5일	35
	레벨1 / 초급	JAVA 기반 문제 해결 기법	40시간 / 5일	36
	레벨2 / 중급	실무 Web Programming	40시간 / 5일	37

실무에서 필요한 필수
과목들로만 구성된
데이터 과학
분야

Data Science 분야

* 과목명을 누르면 과목 상세 소개 페이지로 이동합니다

수준	과목명	표준 시간	페이지
레벨1 / 초급	바로 실무에 써먹을 수 있는 SQL	40시간 / 5일	39
레벨2 / 중급	Advanced SQL 활용 실무	40시간 / 5일	40
레벨2 / 중급	SQL Tuning 실무	40시간 / 5일	41
레벨2 / 중급	데이터 분석을 위한 확률, 통계, 머신러닝	40시간 / 5일	42
레벨1 / 초급	Xlwings를 이용 엑셀 업무 자동화	40시간 / 5일	43
레벨2 / 중급	Pandas를 활용한 데이터처리	40시간 / 5일	44
레벨2 / 중급	Python 모듈을 이용한 데이터 시각화	40시간 / 5일	45
레벨2 / 중급	Python 모듈을 이용한 Dash Board 구현	40시간 / 5일	46
레벨2 / 중급	PyQt를 이용한 GUI Programming	40시간 / 5일	47

빠르게 변하는 AI에
맞춘 발 빠른 과정
업그레이드

인공지능 분야

* 과목명을 누르면 과목 상세 소개 페이지로 이동합니다

수준	과목명	표준 시간	페이지
레벨1 / 초급	Python으로 배우는 통계학	40시간 / 5일	49
레벨2 / 중급	인공지능을 위한 선형 대수학	40시간 / 5일	50
레벨2 / 중급	Scikit-learn을 활용한 머신러닝의 이해와 구현	40시간 / 5일	51
레벨2 / 중급	Keras 기반 딥러닝의 이해와 구현	40시간 / 5일	52
레벨2 / 중급	PyTorch 기반 딥러닝의 이해와 구현	40시간 / 5일	53
레벨3 / 고급	Deep Learning 심화	40시간 / 5일	54
레벨4 / 전문	영상처리를 위한 CNN 실무	40시간 / 5일	55
레벨4 / 전문	시계열 처리를 위한 RNN 실무	40시간 / 5일	56
레벨4 / 전문	생성모델을 위한 GAN 실무	40시간 / 5일	57
레벨4 / 전문	강화학습 실무	40시간 / 5일	58
레벨3 / 고급	임베디드 리눅스 기반 온디바이스 AI	40시간 / 5일	59
레벨3 / 고급	Non-OS 임베디드 시스템 기반 온디바이스 AI	40시간 / 5일	60

SW개발자 및
HW개발 신입사원을
위한 교육 과정

Hardware 설계 분야

* 과목명을 누르면 과목 상세 소개 페이지로 이동합니다

수준	과목명	표준 시간	페이지
레벨1 / 초급	SW 개발자를 위한 디지털 회로 기초	40시간 / 5일	62
레벨1 / 초급	SW 개발자를 위한 임베디드 시스템 설계	40시간 / 5일	63
레벨1 / 초급	Verilog와 FPGA를 이용한 하드웨어 설계 초급	40시간 / 5일	64
레벨2 / 중급	Verilog와 FPGA를 이용한 다양한 Peripheral 설계	40시간 / 5일	65

Embedded System 분야

* 과목명을 누르면 과목 상세 소개 페이지로 이동합니다

수준	과목명	표준 시간	페이지
레벨1 / 초급	임베디드 시스템 필수 디바이스 프로그래밍	40시간 / 5일	67
레벨2 / 중급	임베디드 시스템 확장 디바이스 프로그래밍	40시간 / 5일	68
레벨2 / 중급	이더넷과 WiFi를 이용한 IoT디바이스 설계 실무	40시간 / 5일	69
레벨3 / 고급	Arm Cortex-M 프로세서 시스템 프로그래밍	40시간 / 5일	70
레벨4 / 전문	Arm Cortex-A 멀티코어 시스템 플랫폼 설계	40시간 / 5일	71
레벨2 / 중급	FreeRTOS 커널 포팅 및 멀티태스킹 응용 프로그램 개발	40시간 / 5일	72
레벨4 / 전문	OS 만들면서 배우는 Cortex-A 코프로세서 구조 및 제어	40시간 / 5일	73

초급부터 고급까지
리눅스 개발 경력자가
진행하는 실무
교육

Embedded Linux 분야

* 과목명을 누르면 과목 상세 소개 페이지로 이동합니다

수준	과목명	표준 시간	페이지
레벨1 / 초급	재미있게 배우는 임베디드 리눅스 활용	40시간 / 5일	75
레벨1 / 초급	실전 임베디드 리눅스 Application 프로그래밍	40시간 / 5일	76
레벨1 / 초급	리눅스 명령어 및 셸 프로그래밍	40시간 / 5일	77
레벨2 / 중급	리눅스 IPC의 이해와 응용	40시간 / 5일	78
레벨2 / 중급	임베디드 리눅스 커널 포팅 및 디바이스 드라이버	40시간 / 5일	79
레벨2 / 중급	임베디드 리눅스 멀티미디어 & 네트워크	40시간 / 5일	80
레벨2 / 중급	임베디드 시스템 및 리눅스 파일 시스템의 이해	40시간 / 5일	81
레벨3 / 고급	고급 임베디드 리눅스 디바이스 드라이버 설계	40시간 / 5일	82

Part 2. C언어 Programming 분야 교육 과정

Level 1

C 프로그래밍

선수지식	없음	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	본 과정은 C언어를 처음 배우는 입문자를 위한 기초 문법 및 활용 과정으로 소프트웨어 개발자 양성을 목표로 한다. 따라서 본 과정은 C언어 문법뿐 아니라 구조적인 분석법을 활용하여 C언어로 만들어진 프로그램의 동작원리를 이해하고 분석할 수 있도록 구성되어 있으며 기존 일반 C 언어 교육과 달리 프로그램 관점을 메모리 해석과 스택 운영에 대한 지식까지 배양하여 추후 프로그램 개발 업무시에 보다 신뢰성 있는 프로그램을 개발할 수 있는 기본 능력을 배양한다. 특히, CODEXPERT 온라인 자동 채점 시스템을 사용하여 많은 코드 구현 문제들을 Quiz를 풀듯이 코드를 작성하면서 자연스럽게 C언어 기반의 코드 구현 기법을 익힐 수 있게 된다. 본 과정은 대학에서 배우는 기초 C언어 수준으로 진행되며 한번도 C언어 문법 수업을 수강한 적이 없는 입문자에게 추천하는 과정임.				
교육목표	• C언어 기초 문법 확립 • 프로그램 작성을 위한 기초적인 코드 설계 능력 배양 • 프로그램의 동작 원리의 이해 능력 제고				
교육내용	• 상수와 메모리 표현 • 변수 선언과 Storage Class, Scope, Life Time, Linkage의 개념 • 손파일링 기법(스택 동작에 의한 구조적인 프로그램 분석법)을 활용한 코드 분석 • Visual Studio의 디버거를 이용한 디버깅 • 연산자 제대로 알기, 비트처리 연산자 • 제어문과 반복문 • 포인터 기본 동작 및 힙 액세스 • 선행처리기 및 조건부 컴파일 • 표준 라이브러리 함수				

Level 1

C 기반 문제 해결 기법

선수지식	C 문법의 기본적인 이해	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	국문법을 잘 안다고 소설을 잘 쓸 수는 없듯이 C 문법을 잘 이해해도 주어진 문제를 해결하기 위한 코드를 구현하는 것은 별개의 문제이다. 따라서 최근의 대학 교육도 교양 프로그래밍 과목에서 문법보다도 Computational Thinking에 더 초점을 둔 강의들을 진행하고 있다. 본 과정은 C언어를 기반으로 주어진 문제를 이해 한 후 어떻게 해결할 것인가를 고민하고 이를 코드로 구현하면서 주어진 문제의 해결 능력을 양성하는 실습위주의 과정으로 6가지의 유형의 문제들을 Quiz를 풀듯이 CODEXPERT 자동 온라인 채점시스템을 이용하여 실전적으로 실습을 수행하여 보다 정확하고 최적화된 코드를 구현할 수 있도록 학습하여 각종 코딩 테스트 기초 수준 문제들을 해결할 수 있는 능력을 배양한다.				
교육목표	• Computational Thinking을 위한 문제 이해 및 문제 해결 방안 도출 능력 제고 • 정확한 문법을 사용하여 적절한 코드를 구현하는 능력의 제고 • 아이디어 도출을 통한 심도있는 문제의 해결 연습				
교육내용	• 입출력 연습 • 단계별 문제해결 연습 • 6가지 유형에 대한 문제 풀이 1 유형. 문자열처리 2 유형. 수식/연산 3 유형. 반복/순차처리 4 유형. 탐색/조건비교 5 유형. 규칙/경우의수 6 유형. 도형/기하/격자 • 복합 유형의 아이디어성 문제 풀이				

Level 2

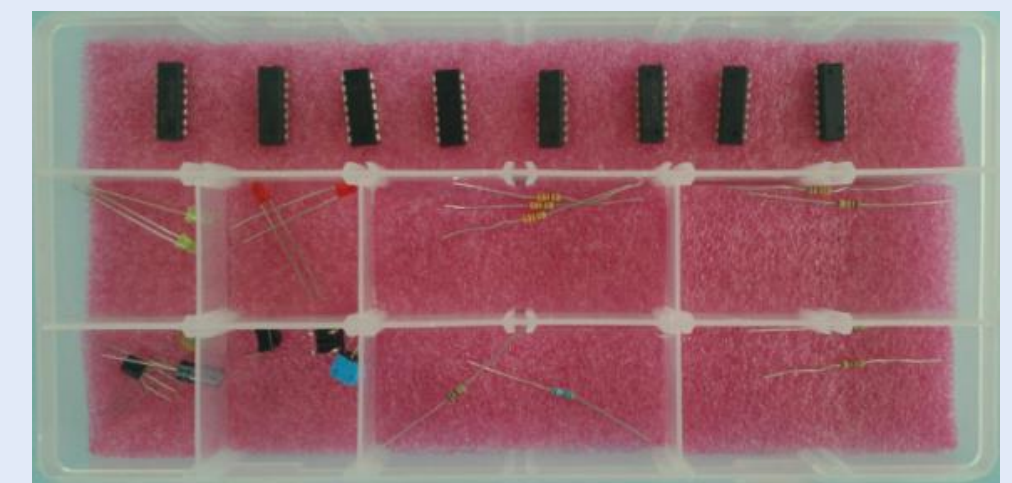
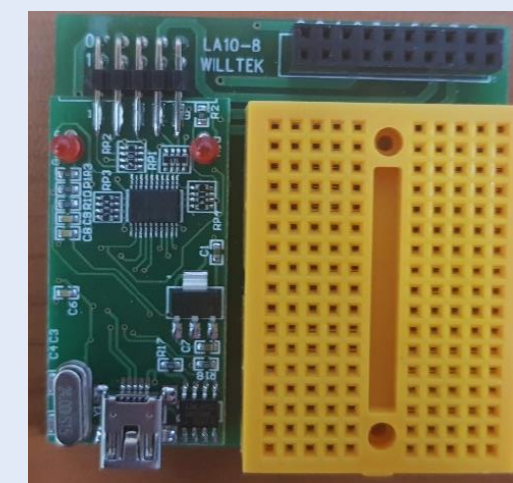
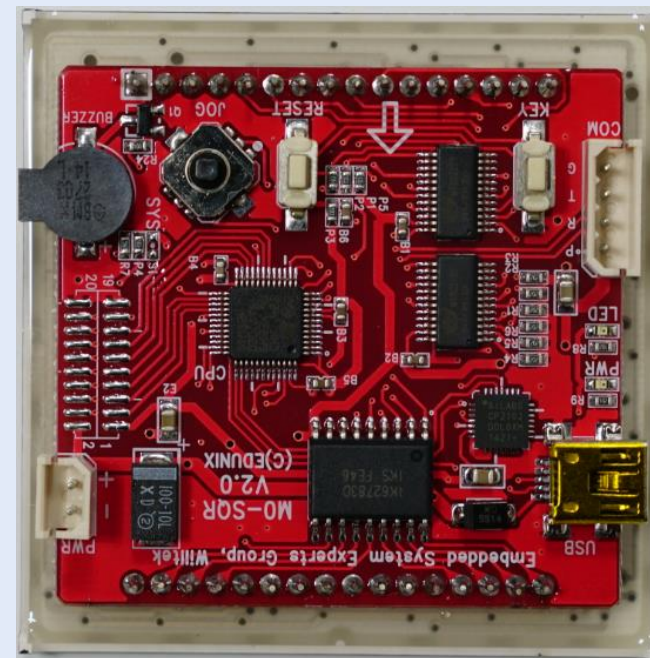
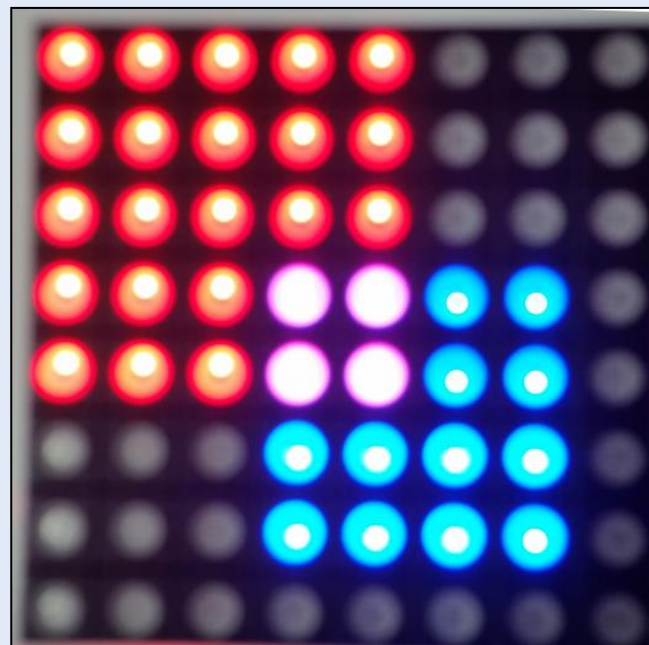
게임을 만들며 배우는 임베디드 C

목표

반도체 소프트웨어 및 임베디드 시스템 프로그래밍의 가장 기초가 되는 C언어를 임베디드 시스템에 활용하기 위한 제반 지식을 습득한다

소개

주어진 부품들을 이용하여 브레드보드에 회로를 구성하고 마이컴 보드와 연결하여 C언어로 제어하는 실습을 진행한다



- ① C언어 개발환경구축, C언어 문법 이해, C언어 문법을 이용한 프로그램 설계 연습(온라인 자동채점 시스템 활용)
- ② 임베디드 시스템을 위한 C언어 활용(비트 연산 처리)
- ③ 디바이스 제어를 위한 제공되는 라이브러리 함수(LED Matrix, Buzzer, Joystick 입력, 그래픽 처리) 동작 분석
- ④ 게임 제작 프로젝트 : 개인들이 아이디어를 도출하여 재미있는 게임 구현 ➔ 경진대회

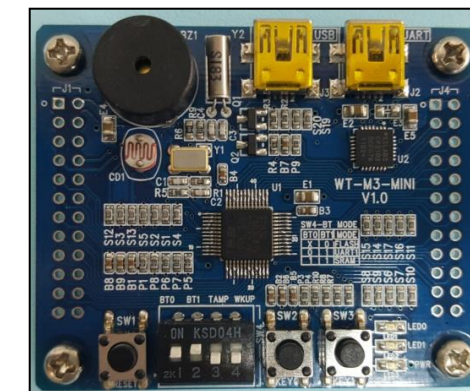
장비

M0-Matrix 보드, CODEXPERT 온라인 자동 채점 시스템

Level 2

실전 임베디드 C 프로그래밍

선수지식	C언어 기초 문법에 대한 이해	강의시간	40시간	실습장비	Arm Cortex-M3 보드
과정개요	signed char x = 0xff; if(x == 0xff) 는 참일까 거짓일까? float f = 3.2; if(f == 3.2) 는 참일까 거짓일까? C언어는 상위 응용 프로그램 개발을 위하여 개발된 JAVA나 C#, C++ 언어들과 달리 OS 및 시스템 제어를 위하여 개발된 언어이기 때문에 원하는 어떠한 기능든지 최적화된 성능으로 구현할 수 있는 장점이 있지만 문법적으로 실수를 하기 쉬운 단점이 있다. 그러나 대부분 SW 개발자들이 대학 수업에서 배운 C언어 지식을 그대로 실무 개발에 사용하고 있는 상황이며 이로 인하여 상당히 신뢰성이 저하되고 최적화 성능이 떨어지는 코드를 구현하는 경우가 많게 된다. 따라서 C언어 개발자들이 실제 제품 개발을 위하여는 보다 정확한 문법의 이해와 활용 능력이 필요하다. 특히, 임베디드 시스템은 PC 기반 개발 환경과 달리 적은 메모리, 낮은 CPU 성능으로 높은 신뢰성을 요구하는 프로그램을 개발하여야 한다. 따라서 임베디드 시스템 프로그램 개발을 위하여 정확한 C언어 문법 지식을 갖추고 이를 정확히 활용할 수 있는 능력을 보유할 필요가 있다.				
교육목표	• C 기반 프로그램 설계를 위하여 반드시 알아야 하는 중요 문법사항의 정립 및 체계화 • 손파일링 기법을 통한 코드의 원론적 분석 능력의 배양 • 메모리 분석을 통한 코드 디버깅 스킬의 배양 • 프로그램에 의한 하드웨어 제어 원리 이해 및 코딩 스킬의 배양				
교육내용	• 임베디드 환경의 이해, 임베디드 개발 환경 구축 • 수와, 메모리, 변수 선언, 정수승격, type conversion, 변수의 Storage Class, Scope, Life Time, Linkage의 개념 • 손파일링 기법(스택 동작에 의한 구조적인 프로그램 분석법)을 활용한 코드 분석 • 하드웨어 제어 : 입출력(GPIO) 제어와 비트연산자 활용, volatile과 const의 활용 • 배열의 활용, 주소처리 연산자, 배열과 포인터, 배열 등가포인터 • Heap Access, malloc의 문제점 • UART 구조와 제어, UART 통신 드라이버 설계, UART용 printf 함수의 설계와 가변인자 전달 • 함수 포인터와 함수 록업 테이블 설계, Key 입력 디바이스 드라이버의 설계 • 선행처리기, 조건부 실행, Assert매크로의 설계				



Level 3

고급 C 포인터 완전정복 및 데이터 모델링

선수지식	C 기반 프로그램 구현 경험 및 능력 보유	강의시간	40시간	실습장비	Arm Cortex-M3 보드
과정개요	포인터를 왜 C언어의 꽃이라 할까? <code>int * (* a[4])(int *)</code>; 이 변수를 해석할 수 있는가? 이 변수의 메모리 구조를 이해할 수 있는가? 아니 이런 코드는 절대 쓸 일이 없다고 생각하는가? 함수 호출 엔진을 만들기 위한 전형적인 함수 Lookup Table을 만드는 변수인데 이것 사용할 일이 없다고 한다면 굳이 프로그램을 만드는데 있어서 C언어를 사용할 필요가 없다. C 언어는 하드웨어 제어부터 상위 응용프로그램 개발에 까지 고르게 사용할 수 있도록 설계된 언어이다. 따라서 시스템 성능 최적화 및 신뢰성 제고를 원한다면 고급 C 포인터를 정확히 사용할 수 있는 능력은 반드시 갖추어야 하는 능력이다. 특히, 리눅스 커널이나 RTOS 커널 등의 분석, 디바이스 드라이버 설계 및 분석 등을 위하여는 더욱 더 필요하다. 특히, 임베디드 시스템 기반의 프로그램 개발의 경우라면 PC 환경과 달리 적은 메모리, 저성능의 프로세서 환경에서 높은 신뢰성을 보장해야 한다. 사람들간 대화하는 영어회화는 초급 수준이라도 자주 사용하면 실력이 늘지만 기계와 대화하는 프로그래밍 언어는 초급 수준의 지식만 가지고 프로그램을 개발하면 평생 초급 프로그램만 개발하게 된다. 즉, 개발 년 수가 증가해도 따라 코드 품질이나 신뢰성이 증가되는 것이 아니라 경험에 의한 코드 설계 시간이 단축되는 정도로만 발전을 하게 된다. 따라서 5년 이상 프로그램 개발하고도 다른 회사에서 개발한 높은 수준의 프로그램이나 OS 커널 코드 등을 제대로 분석하지 못하는 경우도 허다하다. 본 과정은 저자가 고안한 독창적인 고급 포인터 완전정복을 위한 다양한 기법들을 활용한 고급 코드의 설계 및 분석 능력을 배양하여 설계하는 프로그램의 신뢰성 제고와 최적화 성능 구현이 가능하도록 교육한다. 특히 이제는 의미 없어진 전통적인 최적화 기법이 아닌 최신 C 문법에 맞는 최적화 프로그래밍 기법을 함께 학습함으로써 고급 C 기반 프로그램 개발자로 양성하는데 목표를 두고 있다. 또한 학습한 구조체, 공용체, 포인터를 활용한 데이터 모델링 기법을 배우고 이를 활용하여 FAT 파일 시스템의 구현 및 이미지 파일 Parsing을 위한 미니프로젝트를 진행하여 현업 실무 개발 능력을 제고한다.				
교육목표	• 고급 포인터 완전정복 및 신뢰성 제고를 위한 고급 코드의 능숙한 설계 및 직독직해 분석 능력 배양 • 복잡한 고급 코드의 직관적인 분석 및 직관적인 설계 능력의 배양, 메모리와 속도 최적화를 위한 기법 이해 및 활용 능력 배양 • 미니 프로젝트를 통한 데이터 모델링 기법의 이해 및 활용 능력 배양				
교육내용	• 무한 확장형 변수의 해석과 분석 그래프, 주소 처리 연산의 이해와 대치법, 깡통이론, 환산법, 양파이론을 통한 포인터 완전 정복 • 배열 등가포인터, 함수 등가포인터를 활용한 포인터 활용 및 고급 코드 직독직해 • 힙 액세스 및 메모리 직접 액세스, type qualifier(volatile, const)와 최적화, 고급 포인터 타입캐스팅의 활용 • 구조체, 공용체, 비트필드 구조체를 활용한 데이터 모델링 • BMP File Parser, FAT16/32 File System 구현 미니 프로젝트				

Level 3

신뢰성과 최적화를 위한 고급 C 프로그래밍

선수지식	C 기반 프로그램 구현 경험 및 능력 보유	강의시간	40시간	실습장비	Arm Cortex-M3 보드
과정개요	임베디드 시스템은 PC에 비하여 저성능의 CPU, 저용량의 메모리를 사용하며 높은 동작 신뢰성을 요구하고 있다. 따라서 기존에 알고 있는 PC 기반의 프로그램을 설계하기 위한 지식뿐 아니라 최적화되고 신뢰성이 확보되는 코드를 설계하는 능력이 반드시 필요하다. 하지만 기계와 대화하는 C언어는 사람간의 대화와 달리 경력이 늘수록 개발 기간이 단축될 수는 있지만 절대 최적화나 신뢰성을 위한 코딩 능력은 향상되지 않는다. 따라서 5년 이상 프로그램 개발하고도 다른 회사에서 개발한 높은 수준의 프로그램이나 OS 커널 코드 등을 제대로 분석하지 못하는 경우도 많다. 본 과정은 잘못 알고 있는 C언어 문법을 재정립한 후 저자가 고안한 독창적인 다양한 기법들을 학습하여 고급 포인터를 완전정복하고 코드의 신뢰성을 제고하는 코딩 스킬을 학습한다. 이후 C언어, ARM 프로세서, 코프로세서의 특징을 활용한 메모리 및 속도를 최적화하는 기법을 학습한다.				
교육목표	- 고급 코드의 직관적인 분석과 설계가 가능하도록 고급 포인터 및 고급 타입캐스팅 활용 기법을 습득할 수 있다. - 안전한 코드를 설계하기 위한 다양한 코딩 스킬 및 코드 오류를 사전에 방지하기 위한 Assert 기법을 습득할 수 있다. - 스택과 힙 메모리의 누수나 오버플로우를 사전에 탐지하여 런타임 오류를 사전에 대비할 수 있는 기법을 습득할 수 있다.				
교육내용	- 신뢰성 제고를 위한 C언어 문법 재정립 - 신뢰성 있는 C 코드를 작성하기 위한 규칙들 - 코드 리뷰를 위한 고급 포인터 완전 정복 및 직관적 코드 설계 기법 - 고급 포인터 타입캐스팅 및 메모리 직접 액세스 - 신뢰성 제고를 위한 Assert Macro 설계 - 스택과 힙의 오버플로우 관리를 위한 플랫폼 설계 - 데이터 모델링에 의한 코드 신뢰성 향상 및 최적화				

Level 2

C기반 자료구조 프로그래밍

선수지식	C 기반 프로그램 구현 경험 및 능력 보유	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	자료 구조는 효율적으로 데이터를 저장, 검색, 삭제, 활용 하기 위한 체계를 구축하기 위한 방법론이다. 따라서 기존에 잘 알려진 자료구조 기법들을 이해 하고 이를 스스로 코드로 구현하는 능력을 보유함으로써 자신의 프로젝트에 적합한 최적의 자료 구조를 설계할 때 활용할 수가 있게 된다. 특히 자료구조 코드는 대부분 포인터를 활용하고 있기 때문에 자료 구조 기반 코드를 구현하면서 포인터에 대한 실무 활용 능력을 향상 시킬 수가 있다. 또한, 자료구조 는 고급의 문제 해결을 위한 알고리즘 구현시 기본적으로 활용이 되는 지식이므로 고급 알고리즘 구현 능력 확보를 위하여 반드시 습득해야 하는 지식이다. 본 과정은 효율적인 자료를 구성하기 위하여 기존에 알려진 다양한 자료구조들의 동작 원리를 강사의 설명을 통하여 이해하도록 하고 이를 직접 수강자들이 코드로 구현할 수 있도록 하여 자료 구조의 원리와 이해와 포인터 활용 능력을 제고 할 수 있도록 하는 실무 중심으로 진행된다. 특히, 자료구조 관련 도서들이나 타 강의에서 자료구조의 일부 코드를 구현하는 것과 달리 블록을 쌓듯이 코드 기능을 하나씩 완성해 가면서 전체적인 큰 틀의 코드를 설계하도록 실습이 구성되어 있어서 교육 수료 후 실무에 활용할 수 있도록 하였다.				
교육목표	- 가장 기본적인 필수 자료구조의 구조와 동작 원리를 이해하고 이를 C언어로 구현할 수 있는 능력의 배양 - 배열 기반과 링크드 리스트 기반으로 구조체 데이터의 정렬, 탐색, 삽입, 삭제 기능을 구현할 수 있는 능력의 배양 - 데이터 구현을 위한 구조체, 공용체, 배열, 포인터를 활용한 데이터 모델링 기법의 이해 - 효율적인 처리를 위한 이진트리, 해시테이블 등의 데이터 처리 및 재귀호출을 활용한 데이터 처리 능력의 배양				
교육내용	- 배열 기반 자료구조 : 생성, 탐색, 삽입, 삭제 - 다양한 정렬 알고리즘 : Bubble, Selection, Insertion, Quick Sort - 배열 기반 링크드 리스트 : 생성, 탐색, 삽입, 삭제 - 배열 기반 스택, 선형큐, 환형큐 - 힙 기반 링크드 리스트 : 생성, 탐색, 삽입, 삭제, 더블 링크, 다중 링크 - 힙 기반 스택, 큐 - 이진트리 : Traverse, 생성, 탐색, 추가, 삭제, Balanced Tree - 해시 테이블 : 충돌 문제, 공개주소 방식, Chaining 방식				

Level 3

C 기반 Advanced 자료구조 프로그래밍

선수지식	C 기반 프로그램 구현 경험 및 능력 보유	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	자료 구조는 효율적으로 데이터를 저장, 검색, 삭제, 활용 하기 위한 체계를 구축하기 위한 방법론이다. 따라서 기존에 잘 알려진 자료구조 기법들을 이해 하고 이를 스스로 코드로 구현하는 능력을 보유함으로써 자신의 프로젝트에 적합한 최적의 자료 구조를 설계할 때 활용할 수가 있게 된다. 특히 자료구조 코드는 대부분 포인터를 활용하고 있기 때문에 자료 구조 기반 코드를 구현하면서 포인터에 대한 실무 활용 능력을 향상 시킬 수가 있다. 또한, 자료구조 는 고급의 문제 해결을 위한 알고리즘 구현시 기본적으로 활용이 되는 지식이므로 고급 알고리즘 구현 능력 확보를 위하여 반드시 습득해야 하는 지식이다. 본 과정은 기본적인 자료구조 기법을 정리한 후 고급 탐색과 고급 문제해결을 위한 Advanced 자료구조 기법들을 학습함으로써 보다 더 최적화된 자료 관리 프로그램 설계가 가능하도록 한다.				
교육목표	- 가장 기본적인 필수 자료구조의 구조와 동작 원리를 이해하고 이를 C언어로 구현할 수 있는 능력의 배양 - 고급의 탐색을 위한 Advanced Tree 자료구조의 동작 원리를 이해하고 이를 C언어로 구현할 수 있는 능력의 배양 - 고급 문제해결을 위한 Advanced BFS, Advanced DFS 기법의 이해 및 활용할 수 있는 능력의 배양				
교육내용	- Sort(Bubble, Quick, Merge, Heap) - Binary Search - Stack, Queue - Priority Queue - Single Linked List, Double Linked List - Advanced DFS, Advanced BFS - Graph, Hash Table - Segment Tree, Fenwick Tree - Balanced Tree, Red-Black Tree				

Level 3

알고리즘 기반 프로그램 설계 실무

선수지식	C 기반 프로그램 구현 경험 및 능력 보유	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	알고리즘은 컴퓨터로 여러 가지 다양한 문제를 해결하는데 필요한 방법론을 제시한다. 따라서 프로그램 개발은 항상 알고리즘을 개발하는 과정을 수반한다. 이때 특정한 문제에 대한 해결이 필요한 경우는 자신이 직접 해결 방법 즉, 알고리즘을 제시하여 코드를 개발하게 되지만 일반화되고 정형화된 문제에 대한 해결이 필요한 경우는 이미 널리 잘 알려진 알고리즘 기법들을 활용하여 코드를 설계하는 것이 효율적이다. 본 과정은 실무 중심으로 다양한 문제에 대한 해결 방법을 제시하는 가장 필수적이고 보편적으로 널리 사용되는 알고리즘 방법들에 대하여 동작 원리와 실제 코드 구현을 경험하도록 하여 현업 프로그램을 개발할 때 효율적으로 코드를 구현할 수 있는 능력을 양성한다. 특히 온라인 채점시스템을 활용하여 실제 프로그램의 완성도를 평가함으로써 보다 완벽하게 알고리즘을 구현할 수 있는 능력을 배양할 수 있게 된다.				
교육목표	• 난이도 있는 문제 해결을 위한 문제 이해 및 문제 해결 방안 도출 능력 제고 • 프로그램 개발을 위한 널리 알려진 필수 알고리즘들의 원리 이해 및 프로그램 작성시 활용할 수 있는 능력 제고 • 다양한 코딩 테스트 합격을 위한 수준의 실력 양성				
교육내용	• 알고리즘 구현에 활용되는 기법 (재귀호출, 정렬/탐색기법, 필수자료구조) 학습 • 자료구조 기반의 문제 해결 기법 활용 문제 풀이 - 정렬 - 이진 탐색 - 스택, 큐 활용 • 알고리즘 기반의 문제 해결 기법 활용 문제 풀이 - Greedy 기법 - Flood Fill 기법 - DFS 기법 - BFS 기법				

Level 3

알고리즘 기반 프로그램 설계 전문가

선수지식	자료구조, 알고리즘 코드 구현 능력	강의시간	80시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	알고리즘은 컴퓨터로 여러 가지 다양한 문제를 해결하는데 필요한 방법론을 제시한다. 따라서 프로그램 개발은 항상 알고리즘을 개발하는 과정을 수반한다. 이때 특정한 문제에 대한 해결이 필요한 경우는 자신이 직접 해결 방법 즉, 알고리즘을 제시하여 코드를 개발하게 되지만 일반화되고 정형화된 문제에 대한 해결이 필요한 경우는 이미 널리 잘 알려진 알고리즘 기법들을 활용하여 코드를 설계하는 것이 효율적이다. 본 과정은 실무 중심으로 다양한 문제에 대한 해결 방법을 제시하는 가장 필수적이고 보편적으로 널리 사용되는 Advanced 자료구조 기법 및 중급, 고급의 알고리즘 방법들에 대하여 동작 원리와 실제 코드 구현을 경험하도록 하여 현업 프로그램을 개발할 때 효율적으로 코드를 구현할 수 있는 능력을 양성한다. 특히 (주)윌텍에서 운영하는 온라인 채점시스템을 활용하여 실제 프로그램의 완성도를 평가함으로써 보다 완벽하게 알고리즘을 구현할 수 있는 능력을 배양할 수 있게 된다. 본 과정은 C나 C++언어 상관없이 수업 참여가 가능하며 원하는 언어로 실습을 진행할 수 있으나 반드시 기본적인 자료구조와 알고리즘 기반으로 프로그램을 설계할 수 있는 실력을 보유하고 있어야 수업 참여가 가능하다.				
교육목표	• 난이도 있는 문제 해결을 위한 문제 이해 및 문제 해결 방안 도출 능력 제고 • 널리 알려진 프로그램 개발을 위한 필수 알고리즘들의 원리 이해 및 프로그램 작성시 활용할 수 있는 능력 제고 • 중급, 고급의 알고리즘을 활용한 문제를 풀이 할 수 있는 능력 제고 • 대기업 SW역량평가의 고급(Professional) 이상 등급 획득을 위한 코딩 능력의 습득				
교육내용	• 정렬(Merge Sort, Heap Sort) • 기본 자료구조 활용(Linked List, Stack , Binary Search) • Heap & Priority Queue • 특수 형태 트리(Segment Tree, Fenwick Tree) • 최단 거리 알고리즘(Floyd-Warshall , Dijkstra) • Advanced BFS / DFS • Dynamic Programming (Coin Change, LIS, Bit Dynamic) • 실전 문제 풀이 (복합 알고리즘 적용 유형 문제 해결)				

Part 3. Programming 언어 분야 교육 과정

Level 1

C++ & OOP Programming

선수지식	C 문법의 기본적인 이해	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	C++은 C언어로부터 파생된 언어지만 문법적으로 비슷한 내용이 많을 뿐 C언어와는 완전히 다른 언어이다. 즉, C++은 객체지향 프로그래밍을 위한 언어이므로 C++을 제대로 학습하기 위하여는 문법적인 요소뿐 아니라 객체 지향 프로그래밍 자체에 대한 완벽한 이해와 더불어 객체지향 프로그램의 장점을 최대한 살려서 프로그램을 개발할 수 있는 능력의 확보가 절대적으로 필요하다. 본 과정은 기존 C언어 기반으로 개발을 진행하던 개발자들이 C++ 언어를 습득하기 위한 기본 과정으로 C언어에서 발전된 문법과 코드 재사용 관련 문법을 통해 객체지향 프로그래밍을 이해하고 장점을 살리는 프로그램을 개발할 수 있는 능력을 배양하기 위한 내용들을 집중적으로 다룬다. 따라서 C언어 과정들과 중복되는 C++ 문법에 대하여는 다루지 않는 관계로 본 과정을 수강하기 위하여는 가급적 기본적인 C언어의 문법적 지식과 C언어를 사용한 기본적인 프로그램 작성 능력은 갖추고 있을 필요가 있다.				
교육목표	• C언어 기반 개발자가 C++ 언어를 습득하기 위한 기초 지식 및 C에서 발전한 C++ 문법의 확립 • C++을 올바르게 사용하기 위한 객체지향 프로그래밍(OOP) 방법론을 이해하고 이를 올바르게 활용할 수 있는 능력의 배양				
교육내용	• C언어에서 발전된 C++의 Upgrade & New Grammar • Class 관련 문법 • 연산자 오버로딩 • Point 클래스 • String 클래스 디자인 • iostream 클래스 모델링 • 상속의 이해 • 재정의와 가상 함수 • 멤버 포인터를 이용한 가상함수 원리 이해				

Level 2

C++ STL Programming

선수지식	C++ 기본 활용 능력	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	C++ 표준 라이브러리인 STL 에 대해서 깊이 있게 다루는 과정입니다. 본 과정에서는 컨테이너, 알고리즘, 반복자등 STL 기본 요소 뿐 아니라 스마트 포인터, function, chrono, thread 등의 다양한 요소를 배울 수 있습니다. 또한 MODERN C++(C++11/14/17)에서 추가된 새로운 라이브러리도 배울 수 있습니다. 본 과정에서는 STL의 단순한 사용법만 배우는 것이 아니라 주요 핵심 코드를 직접 작성해 보면서 STL 이 가진 설계 철학도 배울 수 있습니다.				
교육목표	• STL 의 설계 철학을 이해 한다. • STL 의 다양한 요소를 활용해서 프로그램을 작성할 수 있다.				
교육내용	• 구현으로 배우는 STL 의 설계 철학 • 컨테이너, 반복자, 알고리즘 • 스마트 포인터 • 스레드 라이브러리 • function, bind, chrono 등 다양한 유틸리티 클래스.				

Level 2

C++ Design Pattern & Refactoring

선수지식	C++ 기본 활용 능력	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	경험 있는 개발자들이 특정 문제를 해결하는 입증된 설계기법을 문서화 시켜 놓은 것이 Pattern입니다. 따라서 패턴을 이해하고 적재적소에 사용함으로써 문제 해결을 위해 고민해야 하는 시간을 줄여서 보다 빠르고 견고하게 품질 좋은 SW를 설계할 수 있습니다. 본 과정에서는 패턴에 대한 내용과 객체지향 기반에서 패턴을 적용하여 설계하는 기법, 그리고 이미 구축된 SW 에 패턴을 적용하고, 성능 코딩 기법등을 통하여 리팩토링을 수행하는 방법을 학습합니다. 본 과정은 C++ 언어를 기반으로 교육이 진행됩니다.				
교육목표	• 디자인 패턴을 이해한다. • 패턴을 적용하여 품질 좋은 SW 설계할 수 있다, • 패턴을 적용하여 SW를 빠르게 설계할 수 있다, • 리팩토링을 통한 견고한 SW를 개발할 수 있다.				
교육내용	• 객체지향이란, 객체지향의 특징, 객체지향 설계 원칙 • 공통성과 가변성의 분리 • 간접층의 원리 • 객체의 생성기술 • 디자인 패턴 카테고리(생성, 구조, 행위)				

Level 3

Modern C++

선수지식	C++ 기반 프로그램 구현 경험 및 능력	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	본 과정은 C++11/14/17 이후에 나온 최신 C++ 기술을 배우고, 최신 경향에 맞는 C++ 프로그램을 개발하는 방법을 배우는 과정입니다. C++11 이후의 최신문법의 등장은 기존의 C++ 코드에 상당히 많은 변화를 가져오게 되었습니다. C++ 기반의 대부분의 오픈소스는 이런 최신의 방식으로 개발 되어가고 있습니다. 본 과정에서는 C++11/14/17 의 문법 뿐 아니라 문법 뒤에 감추어진 설계철학, 최적화를 고려한 고급 기법 등에 대해서 깊이 있게 배우게 됩니다. 대부분의 예제는 안드로이드 커널 등의 오픈소스에서 발췌한 깊이 있는 코드를 사용하여 활용도가 높습니다.				
교육목표	• C++11/14/17 의 최신 문법을 깊이 있는 관점에서 이해 한다 • Perfect forwarding, move semantics, lambda expression, type deduction 등의 핵심 기술을 완벽히 이해하고 활용한다. • 최신 문법을 고려한 새로운 코딩 스타일을 이해하고 활용한다.				
교육내용	• C++11/14/17 에서 추가된 문법 • Perfect forwarding & Move semantics • Lambda Expression 고급 • Type deduction • 오픈 소스가 사용하는 최신 C++ 기술				

Level 1

Python Programming 완벽 이해

선수지식	없음	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	Python은 플랫폼 독립적이며 인터프리터 방식의 객체지향적, 동적 타이핑(dynamically typed) 특성을 갖는 대화형 프로그래밍 언어입니다. Python은 시스템 유틸리티부터 GUI, 웹, 수치연산, 데이터베이스에 이르는 모든 작업이 가능하며, 라이브러리가 풍부하여 교육 기관 및 연구 기관 뿐 아니라 산업 현장에서도 이용이 증가되고 있습니다. 또한 Python은 다른 언어로 쓰인 모듈들을 연결하는 풀 언어(Glue Language)로 사용할 수 있기 때문에 웹 개발, 데이터 베이스, 데이터 처리, 빅데이터, AI분야 등 다양한 많은 응용 프로그램 개발을 위한 스크립트 언어로 Python이 널리 사용되고 있습니다. 흔히 Python 문법은 간단하기 때문에 Python은 문법 학습 보다는 실제 활용을 위한 프로그램 연습 또는 응용을 위한 다양한 모듈의 학습이 더 중요하다고 잘못 알려져 있습니다. 예를 들어 데이터 처리를 하기 위하여 Numpy나 Pandas등에서 제공하는 모듈이나 함수를 잘 사용하는 것도 중요하지만 Python 문법 활용 지식이 반드시 확고하게 기초로 확립이 되어 있어야 자유자재로 응용 프로그램 분석이나 설계가 가능해 집니다. 본 과정은 Python의 문법을 확고하게 이해할 수 있도록 Python의 동작원리와 파이썬을 파이썬답게 사용하기 위하여 중요한 Iterable Type, Comprehension, Lambda Expression을 완벽히 이해할 수 있도록 하며 이를 바탕으로 SW 비직무자들이나 SW 개발 경험이 없는 입문자들을 위하여 Python을 이용하여 올바른 Python 기반 프로그램을 작성할 수 있도록 자체 온라인 채점시스템을 이용하여 프로그래밍 연습을 수행합니다.				
교육목표	• Python 문법을 이해하고 이를 이용한 Python을 Python 답게 사용하는 프로그램 설계 능력 배양 • Namespace, Scope, Binding, Container 구조를 통한 Python 코드의 동작 원리 이해 • 데이터처리, AI등 활용을 위하여 필수적인 Lambda Expression, Iterable Type, Comprehension의 활용 능력 배양				
교육내용	• Python 프로그램 구조, 개발환경 준비, 출력, 입력 함수, 간단한 파일 입출력 활용 • 객체, Namespace, Scope, Binding, 함수 모듈화, 메서드의 개념 이해 및 구조적 프로그램 해석 • 기본 자료형, Container 자료형(tuple, set, list, dict, str)의 생성과 구조, 메서드 활용 • 최적화 및 활용을 위한 Lambda Expression, Comprehension, Iterator의 원리 이해 및 활용 • 연산자, 제어문, 반복문의 동작과 이를 활용한 코드 구현 연습				

Level 1

Python 기반 문제해결 기법

선수지식	Python 기본 활용 능력	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	국문법을 잘 안다고 소설을 잘 쓸 수는 없듯이 C 문법을 잘 이해해도 주어진 문제를 해결하기 위한 코드를 구현하는 것은 별개의 문제이다. 따라서 최근의 대학 교육도 교양 프로그래밍 과목에서 문법보다도 Computational Thinking에 더 초점을 둔 강의들을 진행하고 있다. 본 과정은 C언어를 기반으로 주어진 문제를 이해 한 후 어떻게 해결할 것인가를 고민하고 이를 코드로 구현하면서 주어진 문제의 해결 능력을 양성하는 실습위주의 과정으로 6가지의 유형의 문제들을 Quiz를 풀듯이 CODEXPERT 자동 온라인 채점시스템을 이용하여 실전적으로 실습을 수행하여 보다 정확하고 최적화된 코드를 구현할 수 있도록 학습하여 각종 코딩 테스트 기초 수준 문제들을 해결할 수 있는 능력을 배양한다. 특히 본 과정은 다른 언어의 문제해결 기법 과정과 달리 철저하게 Python다운 코드를 설계하기 위하여 Python의 고급 핵심 기능인 Comprehensions, Lambda, Iterable method 등을 사용하여 간결하고 최적화된 코드를 작성하는 기법을 체득할 수 있게 된다.				
교육목표	• Computational Thinking을 위한 문제 이해 및 문제 해결 방안 도출 능력 제고 • 정확한 문법을 사용하여 적절한 코드를 구현하는 능력의 제고 • 아이디어 도출을 통한 심도있는 문제의 해결 연습				
교육내용	• 입출력 연습 • 단계별 문제해결 연습 • 6가지 유형에 대한 문제 풀이 1 유형. 문자열처리 2 유형. 수식/연산 3 유형. 반복/순차처리 4 유형. 탐색/조건비교 5 유형. 규칙/경우의수 6 유형. 도형/기하/격자 • 복합 유형의 아이디어성 문제 풀이				

Level 2

데이터 처리를 위한 Python 완벽 이해

선수지식	Python 코드 구현 능력	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	Python은 플랫폼 독립적이며 인터프리터 방식의 객체 지향적, 동적 타이핑(dynamically typed) 특성을 갖는 대화형 프로그래밍 언어입니다. Python은 시스템 유틸리티부터 GUI, 웹, 수치 연산, 데이터베이스에 이르는 모든 작업이 가능하며, 라이브러리가 풍부하여 교육 기관 및 연구 기관 뿐 아니라 산업 현장에서도 이용이 증가하고 있습니다. 또한 Python은 다른 언어로 쓰인 모듈들을 연결하는 풀 언어(Glue Language)로 사용할 수 있으므로 웹 개발, 데이터베이스, 데이터 처리, 빅데이터, AI 분야 등 다양한 많은 응용 프로그램 개발을 위한 스크립트 언어로 Python이 널리 사용되고 있습니다. 본 과정은 Python의 문법을 확고하게 이해할 수 있도록 Python의 동작 원리와 파이썬을 파이썬답게 사용하기 위하여 중요한 Iterable Type, Comprehension, Lambda Expression, Class 생성, Exception 처리 등을 완벽히 이해할 수 있도록 하며 이를 바탕으로 비SW 직무자 및 SW 개발 경험 없는 입문자를 위하여 Python을 이용하여 올바른 Python 기반 프로그램을 작성할 수 있도록 자체 온라인 채점 시스템을 이용하여 프로그래밍 연습을 수행합니다. 또한, 데이터 처리에 필수적으로 필요한 File 입출력, 폴더 관리, OS 기능 활용, Windows 시스템 관리, Pickle, Shelves를 이용한 파일 관리, 정규식(re)를 이용한 test 탐색 및 조작 등의 기능을 Python 제공 모듈들을 활용하여 구현하는 방법을 학습합니다.				
교육목표	• Python 문법을 이해하고 이를 이용한 Python을 Python 답게 사용하는 프로그램 설계 능력 배양 • Namespace, Scope, Binding, Container 구조를 통한 Python 코드의 동작 원리 이해 • 데이터처리, AI등 활용을 위하여 필수적인 Lambda Expression, Iterable Type, Comprehension의 활용 능력 배양 • 파일조작, Windows 시스템 조작, Text 탐색 및 조작을 위한 다양한 파이썬 모듈 활용 능력 배양				
교육내용	• 객체, Namespace, Scope, Binding, 함수 모듈화, 메서드의 개념 이해 및 구조적 프로그램 해석 • 기본 자료형, Container 자료형(tuple, set, list, dict, str)의 생성과 구조, 메서드 활용 • 최적화 및 활용을 위한 Lambda Expression, Comprehension, Iterator의 원리 이해 및 활용 • Class 생성, 상속 등의 객체 지향 특성 이해 및 활용 • File 관리와 file 입출력, directory 관리, file 제목 조작, glob 활용, os 모듈을 이용한 windows 시스템 관리, pickle과 shelve를 이용한 파일 관리 • Regular expression을 활용한 text 탐색, 조작				

Level 2

Python 기반 알고리즘 프로그래밍

선수지식	Python 코드 구현 능력	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	알고리즘은 컴퓨터로 여러 가지 다양한 문제를 해결하는데 필요한 방법론을 제시한다. 따라서 프로그램 개발은 항상 알고리즘을 개발하는 과정을 수반한다. 이때 특정한 문제에 대한 해결이 필요한 경우는 자신이 직접 해결 방법 즉, 알고리즘을 제시하여 코드를 개발하게 되지만 일반화되고 정형화된 문제에 대한 해결이 필요한 경우는 이미 널리 잘 알려진 알고리즘 기법들을 활용하여 코드를 설계하는 것이 효율적이다. 본 과정은 실무 중심으로 다양한 문제에 대한 해결 방법을 제시하는 가장 필수적이고 보편적으로 널리 사용되는 알고리즘 방법들에 대하여 동작 원리와 실제 코드 구현을 경험하도록 하여 현업 프로그램을 개발할 때 효율적으로 코드를 구현할 수 있는 능력을 양성한다. 특히 온라인 채점시스템을 활용하여 실제 프로그램의 완성도를 평가함으로써 보다 완벽하게 알고리즘을 구현할 수 있는 능력을 배양할 수 있게 된다. 특히 본 과정은 다른 언어의 문제해결 기법 과정과 달리 철저하게 Python다운 코드를 설계하기 위하여 Python의 고급 핵심 기능인 Comprehensions, Lambda, Iterable method 등을 사용하여 간결하고 최적화된 코드를 작성하는 기법을 체득할 수 있게 된다.				
교육목표	• 난이도 있는 문제 해결을 위한 문제 이해 및 문제 해결 방안 도출 능력 제고 • 프로그램 개발을 위한 널리 알려진 필수 알고리즘들의 원리 이해 및 프로그램 작성시 활용할 수 있는 능력 제고 • 다양한 코딩 테스트 합격을 위한 수준의 실력 양성				
교육내용	• 알고리즘 구현에 활용되는 기법 (재귀호출, 정렬/탐색기법, 필수자료구조) 학습 • 자료구조 기반의 문제 해결 기법 활용 문제 풀이 - 정렬 - 이진 탐색 - 스택, 큐 활용 • 알고리즘 기반의 문제 해결 기법 활용 문제 풀이 - Greedy 기법 - Flood Fill 기법 - DFS 기법 - BFS 기법				

Level 1

C# Programming

선수지식	없음	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	C#은 C++을 기본으로 Visual Basic의 편의성과 최신 언어들의 특징을 결합하여 만든 객체 지향 및 함수 지향형을 지원하는 멀티 패러다임 프로그래밍 언어로 Microsoft에서 닷넷 프레임워크 및 닷넷 코어 플랫폼을 위해 개발한 것이다. 특히 C#은 개발의 생산성, 편의성, 다양성 지원 등의 장점들과 Visual Basic의 사용자 친화성, C++의 객체 지향성, Java의 분산처리 등의 특징을 모두 가지고 있으며, 컴포넌트 기반의 S/W 개발 패러다임을 반영한다. 또한, 윈도우 프로그래밍 및 닷넷 프레임워크 및 닷넷 코어를 지원하고 운영체제와 독립적인 웹 및 응용 프로그램을 만들 수 있는 장점이 있다. 본 과정은 객체 지향 프로그래밍의 기본 단위인 객체 및 메소드를 이용한 프로그래밍을 이용할 수 있도록 하기 위해 관련 개념을 이해한 뒤, 작은 단위의 프로그램 예제를 사용하여 문법 및 구조를 익힌다. 또한, C#의 문법과 객체지향 개념을 이용하여 데이터베이스 프로그램을 작성하여 어떻게 적용되는 지 알 수 있도록 다양한 예제 코드를 살펴보고 직접 흐름을 파악하는 실무 중심의 학습으로 진행된다.				
교육목표	• NET Framework의 이해 • 객체지향과 멀티 스레드를 이용한 프로그램 구현 능력 제공 • Winform과 ADO.net을 이용한 데이터 베이스 프로그램 작성 능력 배양				
교육내용	• NET Framework의 이해, 개발 환경 구축, Visual Studio .NET의 환경 구축, • 데이터 형, 연산자, string, 객체 지향 프로그래밍 개념 잡기, System.Console, 제어문과 배열 • 클래스의 기본 개념, 접근 제한자, 메소드, Overloading, Overriding • main 메서드와 static, 네임스페이스와 클래스 고급, 네임스페이스와 using • 예외 처리,속성, 연산자 오버로딩 • Windows Form, Windows Programming, 필수 이벤트 프로그래밍, 사용자 정의 클래스, 구성요소 클래스 • 델리게이트, 람다식, LINQ, Dynamic • 스레드 이해, 작성, 제어, 동기화				

Level 2

WPF를 이용한 GUI Programming

선수지식	C# 프로그램 이해	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	WPF(Windows Presentation Foundation)는 기존의 GUI제작 기술인 Windows Form이 가지고 있던 문제점인 다양한 미디어 지원, 디자인과 실행과 코드의 분리, 하드웨어 가속, 프로그램 언어에 종속의 문제를 해결하고, 최신 기술 트렌드를 적용하기 위해 도입된 GUI개발 기술이다. WPF는 GUI와 개발언어를 분리하여 개발 언어가 바뀌거나 코드가 변경되어도 GUI를 코드를 변경할 필요가 없다. 모든 언어가 인식할 수 있는 xml로 구현하여 프로그래밍 언어에 독립적이고 프로그래밍에 익숙하지 않은 디자이너도 개발 가능하다. 하드웨어 가속을 이용하여 부드럽고 다양한 화면효과를 지원하고 다양한 사용자 정의 UI 요소를 생성 가능하고, windows 및 web을 동시에 지원 가능한 GUI를 생성할 수 있다. 본 과정은 WPF를 이용하여 사용자 중심의 GUI를 구성하고, 프로그램과 연동 방법을 교육을 통해 이해한 뒤, 작은 단위의 프로그램 예제를 사용하여 문법 및 구조를 익힌다. 또한, MVVM 패턴이 Project에 어떻게 적용될 수 있는지를 다양한 예제 코드로 실습하여 직접 흐름을 파악하는 실무 중심의 학습으로 진행된다.				
교육목표	• WPF(Windows Presentation Foundation)의 특징 및 이해할 수 있다. • C#으로 WPF 기반으로 GUI 응용 프로그램을 개발하는 능력을 배양한다.				
교육내용	• WPF의 특징 및 이해 • WPF Application 생성 및 구조 이해 • 다양한 컨트롤 특징 및 이용 • XAML: 문서구조, 네임스페이스, 객체, 속성, 이벤트, 명명규칙 • Layout: 레이아웃 구조, Canvas, Panel, Grid • Data Binding: Data Context, 변환기, 다양한 컨트롤과 바인딩, INotifyPropertyChanged, INotifyCollectionChanged • 사용자 정의 컨트롤: TemplateBinding, ItemsPresenter, ResourceDictionaries, 변형, 상태, 애니메이션 • MVVM : MVC(Model View Control)의 이해, MVVM의 이해 및 구현				

Level 1

JAVA Programming

선수지식	없음	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	JAVA는 객체 지향 언어의 가장 대표적인 언어이며, 웹 및 임베디드 분야에서 기본적으로 학습해야 하는 언어이다. 특히, 하드웨어에 종속 받지 않는 이식성 높은 언어로 스마트 제품(폰, TV, 냉장고 ...) 에서 사용 가능한 프로그램을 만들 수 있다. 본 과정을 통해 JAVA의 문법을 이해하는 것으로 동적인 웹 프로그래밍 분야에 접근하기 위한 기본 토대를 마련 할 수 있으며, OOP의 기본 단위인 객체와 주요 디자인 패턴 이용한 프로그래밍을 할 수 있도록 하기위해 그 구조 및 자주 사용되는 API들을 학습하고, 작은 단위의 프로그램 예제를 사용하여 문법 및 구조를 익히게 된다. 또한, 다양한 API와 객체지향 개념을 적용하여 데이터베이스 프로그래밍의 부분을 작성하여 실제 Project에서 각 구성요소들과 API들을 어떻게 활용할 수 있는지를 알 수 있도록 하는 실무 중심의 학습으로 진행된다.				
교육목표	• OOP의 기본 구조 및 구성요소, 문법을 이해하고 Java를 이용한 프로그램을 설계하는 능력을 배양한다. • Thread의 개념 및 생성, 종료, 동기화 방법을 학습하여 Multi-Thread 프로그래밍을 할 수 있도록 한다. • JDBC를 이용하여 데이터베이스 프로그래밍을 할 수 있다.				
교육내용	• 자바 기본 문법: 자료형, 형변환, 연산자, 제어문, 배열, reference 개념 • Java의 객체지향 프로그래밍을 위한 문법과 다양한 Class 이해 • 예외처리 • 컬렉션 프레임워크 API: Iterator, Enumeration, Set, List, Map, 제네릭(Generic) • 입출력 API: InputStream, Reader, OutputStream, Writer, DataStream, ObjectOutputStream • Thread: 개념, 생성, 메서드, 종료, 동기화(synchronized 블록, wait()/notify()) • JDBC를 이용한 데이터 베이스 프로그래밍				

Level 1

JAVA 기반 문제 해결 기법

선수지식	JAVA 문법의 기본적인 이해	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	국문법을 잘 안다고 소설을 잘 쓸 수는 없듯이 C 문법을 잘 이해해도 주어진 문제를 해결하기 위한 코드를 구현하는 것은 별개의 문제이다. 따라서 최근의 대학 교육도 교양 프로그래밍 과목에서 문법보다도 Computational Thinking에 더 초점을 둔 강의들을 진행하고 있다. 본 과정은 C언어를 기반으로 주어진 문제를 이해 한 후 어떻게 해결할 것인가를 고민하고 이를 코드로 구현하면서 주어진 문제의 해결 능력을 양성하는 실습위주의 과정으로 6가지의 유형의 문제들을 Quiz를 풀듯이 CODEXPERT 자동 온라인 채점시스템을 이용하여 실전적으로 실습을 수행하여 보다 정확하고 최적화된 코드를 구현할 수 있도록 학습하여 각종 코딩 테스트 기초 수준 문제들을 해결할 수 있는 능력을 배양한다.				
교육목표	• JAVA 기반 프로그램 설계를 위하여 반드시 알아야 하는 중요 문법사항의 재정립 및 체계화 • Computational Thinking을 위한 문제 이해 및 문제 해결 방안 도출 능력 제고 • 정확한 문법을 사용하여 적절한 코드를 구현하는 능력의 제고 • 대기업 SW역량평가의 기본(Intermediate) 등급 획득을 위한 코딩 능력의 습득				
교육내용	• JAVA 문법 정리 • 입출력 연습 • 단계별 문제해결 연습 • 주요 6가지 유형에 대한 알고리즘 문제 풀이 • 1. 문자열처리 • 2. 수식/연산 • 3. 반복/순차처리 • 4. 탐색/조건비교 • 5. 규칙/경우의수 • 6. 도형/기하/격자				

Level 2

실무 Web Programming

선수지식	Java 프로그램 이해	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	Web Programming은 사용자가 접근이 쉬운 웹을 이용하여 서비스를 제공할 때 필요한 기술이다. 데스크탑 부터 모바일까지 한 번의 프로그램으로 지원할 수 있으며, 운영체제와 상관없이 서비스할 수 있고 높은 인기로 인하여 빠르게 발전하고 있다. Web은 다른 기술에 비하여 사용자 인터페이스를 쉽게 구성할 수 있고, 다양한 효과를 사용할 수 있으며, 프로그램 배포 없이 브라우저가 실행 가능한 환경이면 동일한 코드로 PC부터 모바일, TV, 자동차까지 다양하게 지원 가능한 장점이 있다. 본 과정은 웹 프로그래밍으로 프로그램을 작성할 수 있도록 웹 및 각 요소 기술에 대한 기본 구조를 이해하고 특징을 학습한다. 특히 JavaScript의 경우 다양한 프로그램 패러다임(객체지향형, 함수지향형)을 지원하여 Modern Language의 특징과 설계 방식을 경험할 수 있고, 실무에서 적용 가능한 단위 실습으로 확인하고 최신 기술 트렌드에 대해 알아본다. 특히 본 과정에서는 서버 실행환경을 직접 구축하고 java Servlet와 JSP, JDBC등 서버관련 기술의 주요 API를 단위 실습과 미니 프로젝트를 수행하여 학습한다.				
교육목표	• 웹 서비스 및 각 요소 기술의 상호 연동을 이해할 수 있다. • Html5 기반의 Html, JavaScript, CSS등 웹 클라이언트에 관련된 기술을 습득한다. • 서버 실행 환경 설정 및 Java Servlet과 JSP, EL, JSTL 등을 이용하여 웹 서버 프로그램을 할 수 있다. • JDBC를 이용하여 데이터 베이스 연동을 구현하고, MVC 패턴을 이해할 수 있다.				
교육내용	• Web Service Architecture • HTML 5: Tag, 웹 프로그래밍을 위한 구조 설계, 새롭게 추가된 Form, Input 기능을 이용한 사용자 입력 제어. • CSS: CSS와 Rendering, Box model, 다양한 Selector, Property, Value, 구조 설계 • Java Script: 기본 문법, 실행 Architecture, 함수, DOM, BOM, 내장객체, Promise 패턴, Ajax, JQuery, Vue • Tomcat, Mariadb 설치 및 환경 설정 • Servlet의 이해, 생명주기, 입출력, RequestDispatcher, Session, 주요 annotation 등 • JSP: 기본 문법(스크립트 태그, 디렉티브 태그, 액션 태그), 내장객체, 자바 빈즈, session, cookie, JDBC, EL, JSTL등 • 데이터 베이스 연동 미니 프로젝트				

Part 4. Data Science 분야 교육 과정

Level 1

바로 실무에 써먹을 수 있는 SQL

선수지식	없음	강의시간	40시간	실습장비	없음
과정개요	본 과정은 실세계의 각종 정보를 컴퓨터 세계로 끌어 내어 프로그래밍을 위한 데이터 구조를 학습합니다. 핵심적인 DB 모델링 이론정리와 데이터 모델링의 기본 개념 및 단계별 적용 방법을 학습하고, 구축된 데이터를 질의하고 추가, 수정, 삭제하는 방법 등을 학습하며 관계형 DB의 특징과 성능 향상을 위한 인덱스 활용, 스토어드 프로시저, 함수, 커서, 트리거등을 학습합니다. 본 과정은 Oracle SQL 및 PL-SQL을 기반으로 교육이 진행됩니다.				
교육목표	• DB의 개념과 특징을 이해할 수 있다. • 관계형 DB를 조작하는 DML, DDL, DCL을 이해하고 활용할 수 있다. • 조인, 인덱스, 뷰를 이해하고 활용할 수 있다. • 오라클 SQL과 오라클 PL-SQL을 활용할 수 있다.				
교육내용	• DB와 DBMS의 구조 • DB 모델링 방법 • ERD 모델과 DB 분석 • SQL의 구조 • DDL의 활용 (create, alter, drop 등) • DML의 활용 (Select, Insert, Update, Delete) • DCL (Commit, Rollback) • Oracle SQL • Oracle PL-SQL의 개념				

Level 2

Advanced SQL 활용 실무

선수지식	SQL 기본 활용 지식	강의시간	40시간	실습장비	없음
과정개요	본 과정은 효율적이면서도 성능을 높이기 위한 SQL의 작성 방법을 확인하고, 올바른 SQL 작성 방법을 실제 사례를 기준으로 소개합니다. 기초적인 SQL의 작성 방법을 숙지하고 있다는 가정하에 프로젝트 진행 단계에서 반드시 알아야 할 기본 SQL부터 고급 SQL까지 작성 능력을 향상화를 목표로 합니다. 또한 상황에 맞는 조인 및 서브 쿼리의 선택, 분석 함수 등의 사용 방법을 숙지하여 보다 효율적이고 개선된 성능을 보장하게 합니다. 본 과정은 실무 사례를 기반으로 실습을 통해 보다 직관적인 이해 및 응용이 가능합니다.				
교육목표	• 서로 다른 비교연산자를 활용하여 동일한 결과를 반환하는 다양한 SQL 문장을 작성할 수 있다. • 다양한 함수를 이용하여 문제를 해결한다. • Advanced Subquery, Set Operation, 계층쿼리의 개념과 사용법 알기 • Advanced Query(Join, Aggregation)의 다양한 표현 방법 익히기				
교육내용	• 데이터 타입에 따른 SQL 문장 작성시 주의사항 • 실행 계획 분리를 위한 NVL, DECODE의 활용 • ROWNUM을 활용한 TOP-N SQL 작성 및 주의 사항 • RANK, DENSE_RANK, ROW_NUMBER를 활용한 TOP-N Query 작성 • 집계 리포트에 ROLLUP, CUBE, GROUPING SET, PIVOT, UNPIVOT 활용방법 • 상관 서브쿼리를 조인으로 변경하여 성능 향상시키는 방법 • 비교 연산자를 변경하여 성능 향상시키는 방법(예 : OR를 IN으로 처리) • 다중 테이블 데이터 삽입, Merge를 활용하여 성능 향상 • 유효하지 않은 날짜 데이터 검색(데이터 검증을 위해 사용)				

Level 2

SQL Tuning 실무

선수지식	SQL 기본 활용 지식	강의시간	40시간	실습장비	없음
과정개요	SQL 개발을 몇 년씩 해왔더라도 SQL 튜닝이라는 주제 자체가 접근하기에는 쉽지가 않은 것이 사실입니다. SQL 개발 시 성능적 관점에서 개발자가 저 지르는 실수의 패턴이 대부분 한정되어 있습니다. 본 과정에서는 SQL문장을 튜닝 하는데 가장 많이 사용되는 인덱스의 구조 및 유형을 이해하고 인덱스를 어떤 형태로 만들어야 하는지를 학습합니다. 업무의 성격에 따라 사용되어 질 수 있는 조인의 유형과 조인시의 성능을 위해서 사용할 수 있는 방법을 습득합니다. 웹과 같은 온라인성 업무에서 정렬 작업을 대치하여 성능을 높일 수 있는 방법을 배우게 됩니다. 복잡한 SQL을 효율적인 간단한 SQL로 변경하는 방법도 익힐 수 있습니다. Query Transformation에 대한 이해 부족으로 SQL문장 튜닝이 실패하는 경우를 종종 볼 있습니다. 본 과정에서는 Query Transformation에 대한 정확한 개념이해로 SQL 문장의 성능을 향상시키는 방법을 학습합니다.				
교육목표	• SQL 튜닝의 기본 개념을 이해하여 적절한 튜닝 계획을 수립할 수 있다. • 잘못 작성된 SQL을 찾아낼 수 있고, 그 문장을 수정할 수 있다. • 다양한 예제를 통해서 튜닝 기법들이 어떻게 활용되는지 이해하고 적용할 수 있다. • 다양한 튜닝 기법들을 적용하여 SQL 문장의 속도를 향상시킬 수 있다. • Query 변형이 성능에 미치는 영향을 파악할 수 있다.				
교육내용	• Oracle SQL Tuning 개요 • 튜닝도구 / 옵티마이저 실행 계획 분석 • INDEX, JOIN, 서브 쿼리, PAGINATION • 동일 데이터 반복 ACCESS 튜닝, 정렬 작업 튜닝 • 복잡한 SQL을 분석 함수를 활용하여 간단한 SQL로 변경하여 성능 향상 • Query Transformation				

Level 2

데이터 분석을 위한 확률, 통계, 머신러닝

선수지식	파이썬 기본 활용	강의시간	40시간	실습장비	없음
과정개요	최근에는 정해진 규칙이 있는 정형이나 규칙이 없는 비정형 등 다양한 형태로 구성된 빅 데이터와 같은 큰 규모의 데이터에서 유용하거나 의미 있는 본질 (insight)을 찾기 위해 수학, 통계, 경영, 프로그래밍, 머신러닝, 딥러닝 등 다양한 분야의 기술들이 융합하여 사용되고 있습니다. 본 과정은 점점 복잡해지고 있는 대규모 정보들로부터 의미 있는 본질(Insight)을 도출하기 위해 사용되는 시각화, 통계, 머신러닝 등 다양한 방법에 대하여 알아보고, 탐색적 데이터 분석(EDA)을 이용하여 본질 도출에 필요한 특성을 통계적 분석과 시각화를 활용하여 추출한 뒤 데이터를 정제하고 머신러닝으로 모델을 생성, 검증하고 평가하는 방법을 학습합니다. 또한, 확률 및 통계, 회귀, 특성 공간 등에 기반을 두고 있는 다양한 머신러닝(Machine Learning) 알고리즘에 대한 학습을 기반 이론 이해와 다양한 데이터셋(dataset) 을 이용한 실습으로 진행합니다.				
교육목표	• Numpy와 Pandas의 사용 목적 및 개념과 자료구조에 관련된 기능 및 관리 방법, 외부 데이터관리 기능을 습득한다. • 탐색적 데이터 분석을 기반으로 요약 통계, 결측치 등 다양한 데이터 분석 및 처리 방법을 학습한다. • 다양한 시각화에 대한 개념과 특징을 Matplotlib, Seaborn을 이용하여 실습하고 결과를 분석하여 데이터들의 특징을 확인한다. • 머신러닝에 관련된 기본 개념 및 용어에 대해 학습한다. • 다양한 머신러닝 알고리즘에 알아 보고 관련된 수학, 통계 및 확률에 대한 내용을 학습한다.				
교육내용	• Numpy: 개요, ndarray, 인덱싱, 슬라이싱, 연산 • Pandas: 개요, Series, DataFrame, 인덱싱, 슬라이싱, 연산, 데이터 로딩, 저장, 시계열, 기술 통계, 요약 통계 • 시각화: Matplotlib, Seaborn, 시각화 분석 • 전처리(EDA: 탐색적 데이터 분석) • 머신러닝의 개념, 검증, 성능평가 • 머신러닝 알고리즘 : k-최근접이웃, 서포트 벡터머신(SVM), 의사결정트리, 나이브베이즈, 앙상블(배깅, 부스팅, 랜덤포레스트, 보팅, 군집화, 선형/로지스틱회귀, 주성분 분석(PCA) 등				

Level 2

Xlwings를 이용한 엑셀 업무 자동화

선수지식	Python, Pandas 기초 지식	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	Python을 이용하여 Excel 작업을 자동화 할 수 있는 패키지는 많이 있습니다. 그러나, Pandas는 Excel을 직접 제어하지 못하고 처리된 결과를 파일로 저장하기만 하며, openpyxl과 XlsxWriter는 암호화된 파일의 열기는 불가능하며, 매크로 기능도 구현할 수 없습니다 Xlwings는 Excel 자동화를 구현하기 위해 Python 언어를 사용하므로, Excel 매크로 작성을 위한 Visual Basic 언어의 학습 부담을 줄여주고, 데이터 처리나 분석, 플로팅에 관련된 수많은 Python 패키지를 사용할 수 있으며, 다른 패키지에서는 불가능한 사내 보안이 적용된 파일의 열기, 매크로 기능, 사용자 정의 함수(UDF)를 사용한 효율적이고 빠른 작업이 가능하다는 장점을 가지고 있습니다. 본 과정은 Xlwings가 제공하는 Excel Object Model을 위한 객체들을 학습하여 Python 스크립트로 엑셀을 제어하고, 매크로 작성법, Excel을 프론트엔드로 사용하기 위한 Python에서의 컨트롤 사용법, 컨트롤과 매크로의 연결, COM 객체의 사용법 및 Excel에서의 Pandas 활용법과 Excel 차트뿐만 아니라 Matplotlib, Plotly 등의 플로팅 패키지들을 활용하는 케이스별 다양한 실습 예제를 통해 단계적으로 학습하여, 추후 단순히 Excel 자동화만이 아닌 Excel을 Dash Board 처럼 활용할 수 있는 프로그램 작성 능력을 배양할 수 있습니다.				
교육목표	• Xlwings에서 사용하는 Excel Object Model의 특징 및 활용법을 습득한다 • Python을 활용하여 Excel 작업을 자동화 하는 방법을 습득한다 • Excel VBA를 대신하여 Python 언어로 매크로 기능을 구현한다 • 고급 사용자 함수를 정의하여 효율적인 Excel 자동화 작업을 구현한다 • Excel에서 제공되는 컨트롤들을 활용하여 사용자 친화적인 인터페이스를 제공한다				
교육내용	• Excel의 Object Model : Book, Sheet, Range, Chart, Picture, Table, Name • Excel 워크북과 Pandas, numpy를 위한 xlwings의 convert와 option • Python의 다른 패키지에서는 구현 불가능한 Excel 매크로 동작을 위한 xlwings의 Run main과 RunPython 이해 • Excel을 Frontend로 활용하기 위한 양식 컨트롤과 ActiveX 컨트롤 • UDF의 개념과 UDF를 활용한 효율적인 Excel 작업 구현 방법				

Level 2

Pandas를 활용한 데이터처리

선수지식	파이썬 기본 활용	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	Python의 외부 모듈인 Pandas를 이용하면 엑셀로 처리하기 복잡하거나 용량이 크거나 반복적인 업무를 자동화 하는데 매우 편리합니다. 또한 처리 속도도 빨라서 대용량 데이터의 가공, 분석, 통계 집계, 시각화 등의 일반 직장 내 업무를 매우 수월하게 자동화 시킬 수 있습니다. 또한 Pandas는 AI 활용 시 입력 데이터의 전처리에 활용할 수도 있는 아주 유용한 모듈이며 그 활용이 매우 빠르게 증가하고 있습니다. 본 과정은 Python의 데이터 처리 및 분석 모듈과 시각화 모듈을 사용하는 방법 및 데이터를 가공하여 원하는 정보를 만드는 방법을 익히기 위한 내용으로 구성되어 있습니다. 즉, 수집된 데이터를 저장하고 다양한 처리, 분석 모듈에서 활용하기 적합한 형태의 데이터 포맷으로 작성하고, 파일로 저장 및 가져오기를 수행하는 방법을 익히도록 하며 수집된 데이터를 활용하기 편리한 형태로 가공하여 다양한 연산을 적용한 후 원하는 정보를 추출하고 의미를 해석하는 능력을 향상시키고, 나아가 엑셀이나 CSV 파일 기반의 데이터 가공, 처리 자동화 프로그램 구현 할 수 있도록 합니다.				
교육목표	• TXT, CSV, Excel 등의 파일을 읽어서 원하는 형태로 가공, 분석, 집계, 시각화를 수행하는 기법을 습득한다. • 가공된 데이터를 다시 원하는 종류의 다양한 파일로 저장하는 기법을 습득한다. • 매 단원마다 제공되는 실습예제와 미니 프로젝트를 통하여 실무 활용 능력을 배양한다. • 수집한 데이터의 가공, 처리 후 이를 다양한 그래프로 시각화하는 기법을 습득한다.				
교육내용	• Pandas를 위한 numpy 요약 정리 및 기본 연산 • Pandas의 파일 입출력, 기본 자료구조(Series, DataFrame, Index)의 이해 • 기본 Indexing, 기본 연산 및 집계, 고급 indexing, 고급 연산 및 집계 • Pivot, Groupby, Categorical, 시계열 데이터의 활용 • Pandas 제공 시각화를 이용한 데이터 시각화 • 다양한 데이터 가공, 처리 워크샵				

Level 2

Python 모듈을 이용한 데이터 시각화

선수지식	파이썬 기본 활용	강의시간	40시간	실습장비	없음
과정개요	데이터 시각화는 데이터에 내재된 통계적 정보를 시각화 결과물의 형태로 드러내는 것이다. 효과적인 시각화를 위해서는 데이터 성격에 맞는 최적의 그래프를 선택하고 이를 뒷받침하는 소프트웨어를 활용할 필요가 있다. Python의 모듈인 Matplotlib, Seaborn을 이용하면 데이터의 다양한 시각화를 위한 그래프, 차트를 생성할 수가 있으며 Pandas나 Openpyxls을 이용하여 문서에 추가시키거나 엑셀 문서에 추가 시킬 수가 있어 현업 업무에 다양하게 활용이 가능하다. 본 과정에서는 Python의 모듈인 Matplotlib, Seaborn 모듈의 라이브러리에 대하여 집중적으로 학습하고 이를 이용한 데이터의 시각화 기법을 실습을 통하여 익힐 수 있으며 Pandas를 활용한 데이터처리 과정과 연계하여 학습하면 현업 업무에서 매우 유용하게 활용될 수 있다.				
교육목표	• 데이터 분석을 이해하고, 데이터 시각화의 필요성을 인지한다. • 데이터에 맞는 그래프를 파이썬 시각화 패키지를 통하여 구현한다. • 다양한 데이터 시각화 라이브러리를 통해 데이터 분석 결과를 효과적으로 표현할 수 있다. • 실제 데이터를 직접 분석하고 시각화하는 프로세스를 구현함으로써 실무에서 응용할 수 있도록 한다.				
교육내용	• 데이터 시각화의 개념과 필요성 • 기초 통계학의 이해 • Pandas 기초와 탐색적 데이터 분석 • 수량 데이터의 시각화, 데이터 분포의 시각화, 비율 데이터의 시각화 • 정량 변수의 관계 시각화, 시계열 데이터의 시각화, 다변량 데이터 시각화 • Matplotlib, Seaborn, Pandas 등 파이썬 시각화 패키지를 활용한 시각화 실습				

Level 2

Python 모듈을 이용한 Dash Board 구현

선수지식	파이썬 기본 활용	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	데이터 시각화는 데이터가 내포하고 있는 통계적 정보를 시각화 결과물의 형태로 드러내는 것입니다. 효과적인 시각화를 위해서는 데이터 성격에 맞는 최적의 그래프를 선택하고 이를 뒷받침하는 소프트웨어를 활용할 필요가 있습니다. 본 과정은 Python의 모듈인 plotly를 이용하여 데이터의 다양한 시각화를 위한 그래프, 차트를 생성하기 위한 방법들을 학습하며 plotly로 구현한 그래프나 차트 그리고 pandas 및 각종 데이터 처리 모듈들을 이용하여 가공한 데이터를 사용자가 쉽게 활용할 수 있도록 web 기반 사용자 인터페이스를 web browser를 통하여 사용자가 시각화 결과를 확인하거나 다양한 선택 등의 조작을 할 수 있도록 하는 다양한 사용자 인터페이스 모듈들을 지원하는 UI 웹서버를 plotly dash 모듈을 이용하여 구현하는 방법을 학습합니다.				
교육목표	• 데이터 분석을 이해하고, 데이터 시각화의 필요성을 인지할 수 있다 • 데이터에 맞는 그래프를 시각화 패키지를 통하여 구현할 수 있다 • 사용자와 interactive한 대화형 UI를 구현할 수 있다 • 시각화된 결과를 정리하여 web기반 dash board를 구현할 수 있다.				
교육내용	• 개발환경 구축, Dash board구현을 위한 package 소개, Dash 구조이해 • Dash HTML component 이해, Dash Bootstrap 모듈 이해 • Dash Core components 사용, Dash inputs and outputs 이해 • Callback function의 동작 이해 및 callback을 이용한 반응형 web 구현 • Plotly Figure object 이해, Plotly converting figures, Plotly express , Plotly Graphic Object 구현 • Plotly와 Dash의 연동 • Dash Advanced Component 사용(DataTable, Tab, Textaera, ...), Multi page 구현 • 데이터 시각화 및 사용자 인터페이스를 구현하는 미니 프로젝트 수행				

Level 2

PyQt를 이용한 GUI Programming

선수지식	Python기반 프로그램 경험 및 능력	강의시간	40시간	실습장비	CODEXPERT 온라인 자동 채점 시스템
과정개요	PyQt는 C++를 이용한 다양한 플랫폼에서 사용될 수 있는 GUI기반 응용 프로그램 개발 QT를 Python 모듈로 변환해 주는 Framework으로 Qt Designer 프로그램을 이용함으로써 다른 Python 기반 GUI Framework에 비해 더 손쉽게 미려한 UI를 설계할 수 있어서 실무 Python 기반 GUI 프로그램 구현 시 많이 사용되고 있는 외부 모듈입니다. PyQt는 다소 전문 지식이 필요한 C++ 기반이 아닌 Python을 사용하여 GUI를 구현하므로 Python에서 사용할 수 있는 다양한 모듈을 사용하여 AI분야, 빅데이터, 데이터처리, 웹크롤링, 데이터베이스 등등 다양한 분야의 GUI 프로그램을 개발하는 도구로도 활용할 수 있습니다. 본 과정은 PyQt에서 제공하는 기본 widget을 활용하여 GUI 프로그램의 흥미를 유발하고, GUI프로그램의 동작원리인 Signal/Slot, Event를 완벽히 이해 할 수 있도록 하며, 다양한 데이터를 다루기 위한 Model/View Architecture를 이해하고 Python 기반 데이터 처리의 패키지인 Pandas를 위한 Custom Model을 구현한 후 Matplotlib과 연동한 데이터 시각화까지 연계된 실습으로 개별로 존재하던 Python 모듈에 대한 개념을 하나의 프로그램으로 통합할 수 있는 기능을 학습하여 다양한 분야에 응용 할 수 있습니다.				
교육목표	• PyQt의 기본 Widget을 사용한 Python GUI 프로그램의 기본개념 확립 • 다양한 형태의 data를 다루기 위한 Model/View 개념 및 동작원리 이해 • GUI 프로그램에서 시각화 및 데이터 처리를 위한 python모듈 활용 능력 배양				
교육내용	• Widget And Layout • Signal/Slot, Event, User Signal, Windows, Dialog • Model/View : Item View Widget, Standard Model, Custom Model • Pandas, Matplotlib과 연동된 Data 처리 및 시각화 구현 • GUI Application 개발 미니 프로젝트				

Part 5. 인공지능 분야 교육 과정

Level 1

Python으로 배우는 통계학

선수지식	없음	강의시간	40시간	실습장비	없음
과정개요	통계학은 대부분의 학문분야에서 사용되는 사회조사와 실험연구뿐 아니라 국가나 기업의 정책결정 등에 필수적인 역할을 하는 학문으로서 자료를 수집, 정리하고 자료로부터 결론을 도출해내는 방법들을 다루고 있다. 본 과정은 교재만으로 이해하기 어려운 이론 및 증명을 Python을 이용하여 쉽게 정리하며 데이터 분석을 시작할 때 가장 먼저 알아야 하는 통계학적 개념과 이론을 이해할 수 있다. 통계학 전반에 대한 이론적 내용을 학습하고 이를 유용하게 적용시킬 수 있는 응용능력의 배양을 위해, 이론과 실습의 균형 있는 학습을 제공한다. 본 과정을 통하여 데이터가 무엇인지, 어떻게 수집하고 정리하는지 통계학 방법론을 학습하고 Python을 활용하여 실습으로 익힐 수 있게 된다.				
교육목표	• 데이터 분석에 필요한 통계적 개념을 완성 할 수 있다. • 자료의 정리→분석→결론을 내리는데 필요한 통계적 방법론을 마스터 한다. • 더 높은 수준의 통계학 과목을 무리 없이 학습 할 수 있는 기초 실력을 완성한다. • 통계적 모형 - 회귀분석의 개념을 이해하고 모형추정방법, 검정방법, 예측방법에 대해 살펴본다.				
교육내용	• 통계학의 기본 용어 • 기술통계학(Descriptive Statistics) - 데이터의 수집 요약, 정리 • 데이터의 중심위치와 변동의 측정 • 확률과 확률 변수 • 확률분포와 표본 분포 • 추론통계학(Inferential Statistics) - 통계적 가설검정 • t-검정, 카이제곱 검정 등 통계적 검정 • 상관분석 • 통계적 모형 - 회귀분석의 이해				

Level 2

인공지능을 위한 선형 대수학

선수지식	고교 수학(벡터, 행렬, 내적 등) 기초	강의시간	40시간	실습장비	없음
과정개요	선형대수학은 최근에 널리 쓰이는 데이터 분석 및 인공지능 기법에 필요한 핵심적인 수학 이론을 제공한다. 본 과정은 학부 수준의 선형대수학을 주된 내용으로 다루며, 쉬운 예시를 통해 직관적으로 다양한 기초 개념들을 이해할 수 있도록 한다. 또한, 인공지능 태스크에서 이러한 선형대수학 기법들이 다양한 형태로 활용되는 사례를 공부한다. 본 과정을 통하여 인공지능 기술을 깊이 있게 이해하는 데 필요한 선형대수학의 다양한 개념들을 익히고, 해당 인공지능 기술을 다양한 형태로 확장하고 활용할 수 있는 역량을 기르게 된다. 본 과정을 학습하기 위해서는 고등학교 이과 수준의 수학 지식 (벡터, 내적, 행렬, 평면의 방정식 등)을 필요로 한다.				
교육목표	- 인공지능 기술에 필요한 선형 독립, 직교, 최소자승법, 행렬의 특이값 분해 등의 선형대수학의 핵심 개념을 마스터한다. - 더 높은 수준의 선형대수학에 기반한 진보된 기술들을 무리 없이 습득할 수 있는 기초 실력을 완성한다.				
교육내용	- 벡터의 선형 독립, 가우스 소거법 - 선형 함수, 전사, 1대1 매핑 - 부분공간, 행렬의 컬럼 공간 및 널 공간 - 벡터의 직교화, 최소 자승법 - 행렬의 고유값 분해, 특이값 분해 - 딥러닝 및 기타 기계학습 모델에서의 선형대수 기술 활용 사례				

Level 2

Scikit-learn을 활용한 머신러닝 이해와 구현

선수지식	Python, Numpy, Pandas 기초 지식	강의시간	40시간	실습장비	Colab 접속 가능 인터넷 환경 권장
과정개요	본 과정은 데이터 분류와 예측에 필요한 다양한 기계학습 모델과 필요한 기법을 학습하여 파이썬 기반 머신러닝 알고리즘이 어떻게 구현되는지 이해하고 실습하는 과정으로 머신러닝의 기본 이론을 학습하고, Scikit-learn 기반의 실습 위주로 머신러닝을 이해할 수 있습니다. 핵심 머신러닝 개념을 실제 데이터를 활용하여 머신러닝 모형으로 구현해 봅니다. 단, 이 과정을 효과적으로 수강하기 위해서는 파이썬 코드를 이해하는 능력과 파이썬 라이브러리(Numpy, Pandas 등)를 사용해 본 경험이 있거나 코드를 이해할 수 있는 능력이 필요합니다.				
교육목표	- 머신러닝의 다양한 알고리즘 모델들의 핵심적인 개념과 특성을 이해한다. - 파이썬 라이브러리를 활용해 다양한 예시를 활용해 머신러닝 모델을 훈련하고 적용한다. - 다양한 머신러닝 모델을 비교 및 분석하여 데이터에 적합한 기계학습 모델을 선택하고 설정하는 방법을 학습한다.				
교육내용	- 지도학습 / 비지도 학습의 이해 - 데이터 전처리 - 선형회귀를 통한 예측모형의 생성과 모형의 평가 - 과대 적합(overfitting) 문제와 해결 - 분류모형 - 로지스틱 회귀 분석, SVM, Decision Tree, Random Forest - 분류모형의 평가 - confusion matrix, ROC, AUC - 모형의 평가와 선택 - 차원축소 - PCA, t-SNE - 군집분석 - K-means, Hierarchical, DBSCAN - 훈련된 모형의 저장과 복원				

Level 2

Keras 기반 딥러닝의 이해와 구현

선수지식	Python, Numpy, Pandas 기초 지식	강의시간	40시간	실습장비	Colab 접속 가능 인터넷 환경 권장
과정개요	최근 인터넷을 기반으로 한 빅데이터 환경의 조성, 컴퓨터 공학 기술을 기반으로 한 컴퓨팅 파워의 발전, 그리고 1950년대부터 시작되어 온 AI 기술이 머신러닝과 딥러닝을 기반으로 비약적인 발전을 하고 있으며, 다양한 분야에서 활용되고 있습니다. 본 과정에서는 머신러닝과 딥러닝의 기본적인 이론을 학습하고, 이를 바탕으로 하여 Numpy 라이브러리 및 Tensorflow로 해당 내용을 구현하여 그 구조를 이해/활용하는 것을 목적으로 하고 있습니다. 이를 위하여, 머신러닝의 기본적인 Regression/ Classification 모델에 대해 학습하고, 이를 Python 및 Tensorflow로 프로그래밍하여 그 기본적인 개념을 확립한 후, 딥러닝의 시작인 MLP를 직접 구현하고, 실제 다양한 분야에서 활용되고 있는 CNN과 RNN에 대하여 개념/구조를 학습한 후 실제 구현을 통하여 이를 활용할 수 있는 능력을 배양합니다. 추가로 Tensorflow의 활용을 확장할 수 있는 모델 구조, 저장, 복원 및 심화에 필요한 이슈들을 학습합니다.				
교육목표	• A.I. / Machine Learning / Deep Learning에 대한 기본개념을 이해하고, 향후 발전방향을 예측할 수 있는 능력 배양 • 머신러닝의 기본모델인 Regression/Classification을 이해하고, 이를 NumPy / Tensorflow로 구현할 수 있는 능력 배양 • 딥러닝의 기본인 Multi Layer Perceptron을 이해하고, KERAS 기반으로 이를 활용할 수 있는 능력 배양 • 이미지 처리를 위한 CNN의 동작원리를 이해하고, 다양한 입력을 처리하는 모델을 구성/향상 시키는 능력 배양 • 단어 처리를 위한 RNN의 동작원리를 이해하고, embedding을 통한 단어처리 모델을 이해/활용하는 능력 배양 • 구현에 사용되는 다양한 라이브러리들에 대한 이해 및 활용 능력의 배양				
교육내용	• AI, M/L, D/L 이해 • 머신러닝의 이해: Linear Regression, Logistic Regression • 딥러닝으로의 확장: Single Layer Perceptron, Multi Layer Perceptron, keras 모델 • 머신러닝 딥러닝에 사용되는 라이브러리의 활용 • CNN 이해 및 구현 실습 • RNN 이해 및 구현 실습				

Level 2

PyTorch 기반 딥러닝의 이해와 구현

선수지식	Python, Numpy, Pandas 기초 지식	강의시간	40시간	실습장비	Colab 접속 가능 인터넷 환경 권장
과정개요	이 과정을 통하여 딥러닝 학습을 시작하기 위해 알아야 할 기초 개념을 이해하고 예제로 학습합니다. 이 과정을 통하여 딥러닝 학습을 시작하기 위해 알아야 할 기초 개념을 이해하고 실제 데이터로 실습합니다. 본 과정을 통하여 개발자로서의 역량을 키우고 딥러닝, 인공지능 분야에 활용할 수 있는 기반을 마련합니다. 본 과정은 딥러닝의 기본 원리와 이론적 배경을 이해하고 바로 실습해 보는 체감형 학습 방법으로 진행되며 과정을 효과적으로 수강하기 위해서는 파이썬 스크립트에 대한 이해와 파이썬 라이브러리(Numpy, Pandas 등)의 사용 경험이 사전 지식으로 필요합니다.				
교육목표	• 인공 신경망이 학습을 수행하는 원리와 구현방법을 이해할 수 있다. • 딥러닝의 학습모델에 영향을 미치는 요소를 이해할 수 있다. • 딥러닝 모델을 데이터에 맞게 구성하고 적용할 수 있다. • Pytorch 딥러닝 라이브러리에 연동해 모델을 구성하고 훈련할 수 있다. • 최적화/정규화/하이퍼파라미터 튜닝을 이해하고 성능 향상을 위한 방법들을 실습한다.				
교육내용	• 딥러닝 라이브러리인 Pytorch의 기초와 활용 • 딥러닝의 이론적 기본 개념과 Perceptron, DNN(Deep Neural Network) 구현 • 과대 적합(overfitting) 문제와 해결 • 하이퍼파라미터(hyper parameter) 튜닝 • CNN(Convolutional Neural Network)을 활용한 이미지 분류기 구현 • CNN 활용한 Style Transfer의 구현 • Transfer Learning의 이해와 구현 • 시퀀스 데이터를 위한 RNN • RNN, LSTM, GRU 모형의 이해 • Char-RNN을 활용한 text generator의 구현				

Level 3

Deep Learning 심화

선수지식	Python, Numpy, Pandas 기초 지식	강의시간	40시간	실습장비	Colab 접속 가능 인터넷 환경 권장
과정개요	최근 수년간 딥러닝은 획기적인 발전과 함께, 다양한 분야에서 가장 주목 받는 기술을 넘어 최대한 활용해야 하는 기술이 되었습니다. 본 과정은 딥러닝을 실전에 적용할 수 있는 기반을 닦는 것을 목표로, 딥러닝 개발에 필요한 주요 개념을 정리하고, CNN, RNN과 GAN을 중심으로 대표적인 모델들을 이론과 실습을 통해 학습합니다. 영상처리에 많이 사용되는 CNN모델은 AlexNet, VGGNet, GoogLeNet(Inception V1/2/3/4), ResNet, WideResNet, DenseNet, PyramidNet등의 구조와 동작 및 특징점을 학습합니다. 또한 시계열 처리에 많이 사용되는 RNN모델은 기본이 되는 LSTM, GRU구조부터, Multi Layer, Bidirectional, Parallel RNN구조를 학습합니다. 실습을 통해 Char-RNN 및 다양하게 변형된 시계열 모델을 학습합니다. 생성모델 학습을 위해 AutoEncoder, Variational AutoEncoder, basic GAN, DC-GAN 모델을 다루며 모델의 성능을 높이기 위한 최적화 기법들에 대하여 학습하며 실습을 통해 그 한계와 실무 적용가능성을 확인합니다.				
교육목표	• PyTorch / Tensorflow 환경에서 딥러닝 모델 개발 방법 습득 • 모델 최적화를 위한 다양한 기법들의 이론적 배경을 학습하고, 실습을 통해 성능과 한계를 이해 • CNN의 기본구조를 이해하고, 대표적인 Architecture 분석을 통해 적용된 기반 기술을 이해 • RNN의 기본모듈구조를 이해하고, 대표적인 Architecture 분석을 통해 다양한 구조의 모델 설계 방식을 이해 • GAN으로 대표되는 생성모델의 이론적 토대를 학습하고, 실습을 통해 실무 적용 가능성을 확인				
교육내용	• 딥러닝 주요 개념 정리 (ANN, Deep ANN, Layer, Node, Features, Representation, Optimization) • Optimization : Regularization, Batch Normalization, Dropout, Learning rate scheduling, Bayesian optimization • CNN : AlexNet, VGGNet, GoogLeNet, ResNet, WideResNet, DenseNet, Transfer Learning, Class Activation Map • RNN : GRU 및 LSTM 모듈, Stacked LSTM, Parallel LSTM, Bidirectional LSTM, LSTM AutoEncoder, Char-RNN • GAN : AutoEncoder, Variational AutoEncoder, basic GAN, DC-GAN				

Level 4

영상처리를 위한 CNN 실무

선수지식	Python, Tensorflow 활용 가능자	강의시간	40시간	실습장비	Colab 접속 가능 인터넷 환경 권장
과정개요	본 과정은 영상처리분야에 딥러닝 적용을 목표로 합니다. 딥러닝에 관한 초,중급 개념을 빠르게 정리하고, CNN의 기본구조 부터 최신 모델까지 다양한 모델의 구조와 특징을 학습합니다. 대표적인 모델에 대한 이론뿐 아니라 실습 코드를 통해 실무 적용 가능성을 스스로 평가할 수 있도록 합니다. 본 과정은 영상처리에 기본이 되는 주요 CNN모델 AlexNet, VGGNet, GoogLeNet(Inception), ResNet, DenseNet, MobileNet, Squeezenet 등의 구조와 동작 및 특징점을 학습합니다. 또한 기존 학습된 모델을 활용하기 위한 전이학습, 모델 검증에 필수적인 Heat Map 생성과 활용을 학습합니다. 본격적인 응용 분야에 해당하는 영상에서 객체 검출, 영역분리, 영상개선 분야에 사용되는 모델들의 구조와 동작을 학습하며 각 분야에서 최근에 선보이는 기술들을 검토하는 과정입니다.				
교육목표	• Keras환경에서 딥러닝 모델 개발 방법 습득 • 모델 최적화를 위한 주요 기법들의 이론적 배경을 학습하고, 성능과 한계를 이해 • CNN의 기본구조를 이해하고, 대표적인 Architecture 분석을 통해 기반 기술을 이해 • CNN응용을 위한 전이학습, CAM(heat map) 이해 • Object detection 모델들의 구조를 이해하고, 최근 모델 학습 • Semantic segmentation 모델구조를 이해하고, 최근 모델 학습 • 화질 개선에 사용되는 모델들의 구조를 이해하고, 최근 모델 학습				
교육내용	• 딥러닝 개념 정리 : ANN, Deep ANN, Layer, Node, Features, Representation, Optimization • Optimization : Regularization, Batch Normalization, Dropout, Learning rate scheduling, Bayesian optimization • 주요 CNN 모델 : AlexNet, VGGNet, GoogLeNet(V1,2,3,4), ResNet, WideResNet, DenseNet, MobileNet, Squeezenet • CNN 응용 : Transfer Learning, Class Activation Map • Object Detection, Localization : R-CNN, Fast RCNN, Faster RCNN, SPP Net, YOLO1/2/3/4, SSD, EfficientDet • Semantic Segmentaion : Deconvolutional Network, U-Net, Deep Lab, • Image Reconstruction : Super-Resolution, Denoising • 영상처리를 위한 Transformer의 이해와 활용				

Level 4

시계열 처리를 위한 RNN 실무

선수지식	Python, Tensorflow 활용 가능자	강의시간	40시간	실습장비	Colab 접속 가능 인터넷 환경 권장
과정개요	본 과정은 시계열 데이터에 딥러닝 적용을 목표로 합니다. 딥러닝에 관한 초,중급 개념을 빠르게 정리하고, RNN의 기본구조 부터 최신 모델까지 다양한 모델의 구조와 특징을 학습합니다. 대표적인 모델에 대한 이론뿐 아니라 실습 코드를 통해 실무 적용 가능성을 스스로 평가할 수 있도록 합니다. 최근 수 년간 딥러닝은 획기적인 발전과 함께, 다양한 분야에서 가장 주목 받는 기술을 넘어 최대한 활용해야 하는 기술이 되었습니다. 본 과정을 통해 실무적용에 빠르게 접근할 수 있습니다. 시계열 및 자연어처리에 기본이 되는 RNN 모델에 사용되는 LSTM,GRU 구조를 학습 합니다. Parallel LSTM, Stacked LSTM, Bidirectional LSTM을 다양한 형태로 설계하고, 내부 동작을 학습 합니다. 시계열 분석에 전통적으로 사용하였던 SARIMA모델+RNN, CNN+RNN 모델을 통해 확장 성을 학습합니다. 보다 복잡한 구조로 LSTM AE, LSTM SAE, Wavelet + SAE + LSTM, Seq2Seq 등을 학습 합니다. 최근에 각광을 받고 있는 Attention 모델들의 구조와 동작을 학습 합니다.				
교육목표	• Keras환경에서 딥러닝 모델 개발 방법 습득, 모델 최적화를 위한 주요 기법 이해 • RNN Cell의 기본구조를 학습하고, 대표적인 RNN Architecture 설계 및 동작 분석을 통해 기반 기술 이해 • 다른 모델과 결합구조를 학습하기 위해 ARIMA + RNN, CNN + RNN 모델을 학습 • 정보 압축을 위한 LSTM AE(Auto encoder), Seq2Seq 모델을 학습 • Attention 개념을 학습하고, 다양한 Attention모델을 학습				
교육내용	• Time series model : ARIMA, SARIMA • Cell 구조 : RNN, LSTM, GRU • RNN 기본 구조 : Parallel, Stacked, Bidirectional model 설계 (Stateful, Return_State, Returen_Sequence) • RNN 응용 구조 : Multivariate Multistep LSTM, ARIMA + LSTM, CNN + LSTM, • 정보변환을 위한 구조 : LSTM AE, LSTM SAE, Seq2Seq, Seq2Seq Conv • Attention : RNN + Attention, Self-Attention, Multi head self-attention, Dual-stage Attention LSTM • 시계열 처리를 위한 Transformer의 이해와 활용				

Level 4

생성모델을 위한 GAN 실무

선수지식	Python, Tensorflow 활용 가능자	강의시간	40시간	실습장비	Colab 접속 가능 인터넷 환경 권장
과정개요	본 과정은 데이터 생성에 딥러닝 적용을 목표로 합니다. 딥러닝에 관한 초,중급 개념을 빠르게 정리하고, 생성모델의 기초가 되는 Auto Encoder에서 시작하여 다양한 GAN 모델에 대한 구조 및 동작을 학습 합니다. 대표적인 모델에 대한 이론뿐 아니라 실습 코드를 통해 실무 적용 가능성을 스스로 평가할 수 있도록 합니다. 최근 수년간 딥러닝은 획기적인 발전과 함께, 다양한 분야에서 가장 주목 받는 기술을 넘어 최대한 활용해야 하는 기술이 되었습니다. 본 과정을 통해 실무적용에 빠르게 접근할 수 있습니다. 생성모델을 통해 복잡한 데이터를 자동으로 만들어 내는 다양한 GAN 모델을 학습 합니다. 최신 GAN 모델은 인간의 얼굴 이미지를 실물사진 수준으로 생성 합니다. GAN 모델의 발전 단계에 따라 학습하여 복잡한 최근 모델까지 구조와 동작을 이해할 수 있도록 합니다. 최대한 많은 모델을 구현, 동작하여 실무에 활용할 수 있도록 합니다.				
교육목표	• Keras환경에서 딥러닝 모델 개발 방법 습득, 모델 최적화를 위한 주요 기법 이해 • 생성 모델의 기본이 되는 Auto Encoder, Variational Auto Encoder 모델 이해 • GAN의 기본 모델이 되는 Basic GAN, DCGAN, ConditionalGAN, ACGAN, InfoGan 모델 이해 • 고급 이미지 생성을 위한 GAN 모델인 EnhanceNet, SimGAN, Pix2Pix, CycleGAN(DiscoGAN) 모델 이해 • 최신 GAN 모델인 startGAN, styleGAN 이해				
교육내용	• 생성모델 : Auto Encoder, Variational Auto Encoder, basic GAN • 기본 GAN : DCGAN, ACGAN, InfoGan • 고급 GAN(1) : EnhanceNet, SimGAN, Pix2Pix, CycleGAN(DiscoGAN) • 고급 GAN(2) : StarGAN, StyleGAN • 생성모델을 위한 Transformer의 이해와 활용				

Level 4

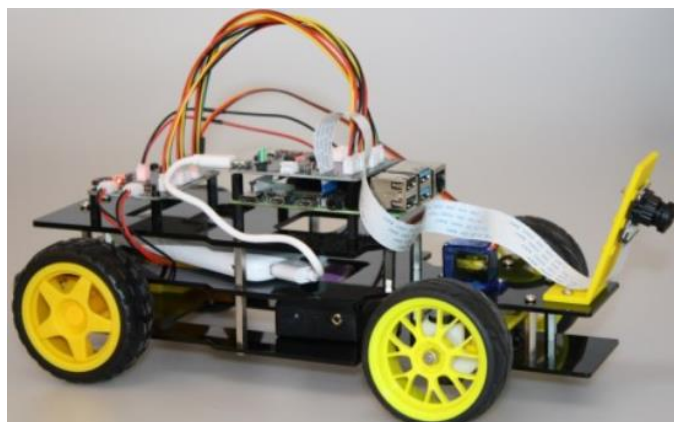
강화학습 실무

선수지식	Python, Tensorflow 활용 가능자	강의시간	40시간	실습장비	Colab 접속 가능 인터넷 환경 권장
과정개요	본 과정은 강화학습에 딥러닝 적용을 목표로 합니다. 딥러닝에 관한 초,중급 개념을 빠르게 정리하고, 강화학습의 기초가 되는 MDP(Markov Decision Process)와 Bellman Equation을 학습합니다. Dynamic Program을 통해 Bellman Equation을 통해 MDP를 해결하는 과정을 학습 합니다. 이러한 접근의 한계를 이해하고, 강화학습을 통한 근사(Monte-Carlo Approximation)를 통해 가치와 정책을 추론하는 과정을 학습합니다. 강화학습의 기본 구조인 SARSA와 Q-Learning 알고리즘을 학습하고, 그 한계를 이해 합니다. 딥-강화학습은 강화학습의 한계를 극복하여, 복잡한 환경에 적용 가능하도록 합니다. 기본 모델로 Deep SARSA와 DQN(Deep Q-Network)모델을 학습 합니다. 그 밖에 속도와 성능이 개선된 다양한 딥-강화학습 알고리즘과 모델들을 학습합니다. 이론뿐 아니라 모델 구현과 실습과정을 통해, 또한 다양한 적용 사례를 통해 실무 적용 가능성을 높이도록 합니다. 딥러닝의 여러 분야들이 강화학습과 연계하여 성능을 크게 개선하고 있기에 딥-강화학습은 딥러닝의 필수 분야가 되었고, 이제 강화학습은 딥-강화학습을 의미하게 되었습니다. 최근 수년간 딥러닝은 획기적인 발전과 함께, 다양한 분야에서 가장 주목 받는 기술을 넘어 최대한 활용해야 하는 기술이 되었습니다. 본 과정을 통해 실무적용에 빠르게 접근할 수 있습니다.				
교육목표	• Keras환경에서 딥러닝 모델 개발 방법 습득, 모델 최적화를 위한 주요 기법 이해 • MDP(Markov Decision Process)와 Bellman Equation 이해 • Dynamic Program을 통한 MDP 해결 과정 학습 • 강화학습 기본구조인 SARSA와 Q-Learning 알고리즘의 동작과 한계 이해 • Deep SARSA와 Deep Q-Network 기본 이해 • 다양한 DQN 모델들의 특성 이해				
교육내용	• 딥러닝 개념 정리 : 딥러닝 개요, 용어 정리, Keras Workflow 핵심 정리, 주요 Layer API 정리 • MDP(Markov Decision Process), Bellman Equation • Dynamic Programming : Policy iteration, Value iteration • Reinforcement Learning : Monte-Carlo Approximation, Temporal-Difference, SARSA, Q-Learning • Deep RL : Deep SARSA, Policy Gradient, DQN, Double Q-Learning, Actor-Critic, A3C • 강화학습을 위한 Transformer의 이해와 활용				

Level 3

임베디드 리눅스 기반 온디바이스 AI

선수지식	Python, Tensorflow 활용 지식	강의시간	40시간	실습장비	WT2837 라즈베리 기반 보드
과정개요	클라우드 서버 또는 성능이 우수한 PC의 GPU, NPU 환경을 이용하여 개발되는 인공지능 기술을 네트워크 접속이나 고성능의 GPU, NPU가 없더라도 프로세서가 탑재된 제품 자체에서 인공지능이 구동되는 온디바이스 AI 기술은 그 활용 범위와 파급력이 매우 크지만 아직까지 제대로 교육이 진행되고 있지 않는 실정입니다. 본 교육은 Tensorflow를 기반으로 클라우드 서버 또는 고성능 GPU 기반 PC 환경에서 훈련된 모델 결과를 Tensorflow Lite 모듈을 이용하여 임베디드 리눅스가 탑재된 시스템에 탑재하여 인공지능 서비스를 구현할 수 있는 기술을 학습하는 과정입니다.				
교육목표	• 딥러닝의 기본인 Multi Layer Perceptron을 이해하고, KERAS 기반으로 이를 활용할 수 있는 능력을 확보할 수 있다. • 이미지 처리를 위한 CNN의 동작원리를 이해하고, 다양한 입력을 처리하는 모델을 구성 / 향상 시키는 능력을 확보할 수 있다. • 인공지능 모델에 의하여 훈련된 결과를 Tensorflow Lite를 이용하여 리눅스 기반 임베디드 시스템에 이식하는 과정을 이해할 수 있다. • 임베디드 시스템의 성능 제약에 따른 경량화 및 최적화에 대한 경험을 습득할 수 있다.				
교육내용	• Tensorflow 기반 딥러닝 구현 기술 정리 • CNN(Convolutional Neural Network)을 활용한 이미지 분류기 구현 • 임베디드 리눅스 기반 응용 프로그램 개발 • Tensorflow Lite를 이용하여 임베디드 리눅스 기반 보드에 이식 • 미니 프로젝트 : 주어진 인공지능 서비스 개발 및 최적화 수행				



Level 3

Non-OS 임베디드 시스템 기반 온디바이스 AI

선수지식	Python, Tensorflow 활용 지식	강의시간	40시간	실습장비	Cortex-M7-ULTRA 보드
과정개요	클라우드 서버 또는 성능이 우수한 PC의 GPU, NPU 환경을 이용하여 개발되는 인공지능 기술을 네트워크 접속이나 고성능의 GPU, NPU가 없더라도 프로세서가 탑재된 제품 자체에서 인공지능이 구동되는 온디바이스 AI 기술은 그 활용 범위와 파급력이 매우 크지만 아직 제대로 교육이 진행되고 있지않는 실정입니다. 본 교육은 Tensorflow를 기반으로 클라우드 서버 또는 고성능 GPU 기반 PC 환경에서 훈련된 모델 결과를 Tensorflow Lite 모듈을 이용하여 C언어 기반으로 동작하는 OS가 탑재되지 않은 임베디드 시스템에 이식하여 인공지능 서비스를 구현할 수 있는 기술을 학습하는 과정입니다.				
교육목표	• 딥러닝의 기본인 Multi Layer Perceptron을 이해하고, KERAS 기반으로 이를 활용할 수 있는 능력을 확보할 수 있다. • 이미지 처리를 위한 CNN의 동작원리를 이해하고, 다양한 입력을 처리하는 모델을 구성/향상 시키는 능력을 확보할 수 있다. • 인공지능 모델에 의하여 훈련된 결과를 C언어 기반으로 동작할 수 있도록 변환한 후 ARM 프로세서 보드에 이식하는 과정을 이해할 수 있다. • 임베디드 시스템의 성능 제약에 따른 경량화 및 최적화에 대한 경험을 습득할 수 있다.				
교육내용	• Tensorflow 기반 딥러닝 구현 기술 정리 • CNN(Convolutional Neural Network)을 활용한 이미지 분류기 구현 • ARM Cortex-M7 주변장치 제어 라이브러리 동작 이해 및 응용 프로그램 구현 • C언어 기반으로 동작할 수 있도록 변환한 결과를 ARM Cortex-M7 보드에 이식 • 미니 프로젝트 : 주어진 인공지능 서비스 개발 및 최적화 수행				

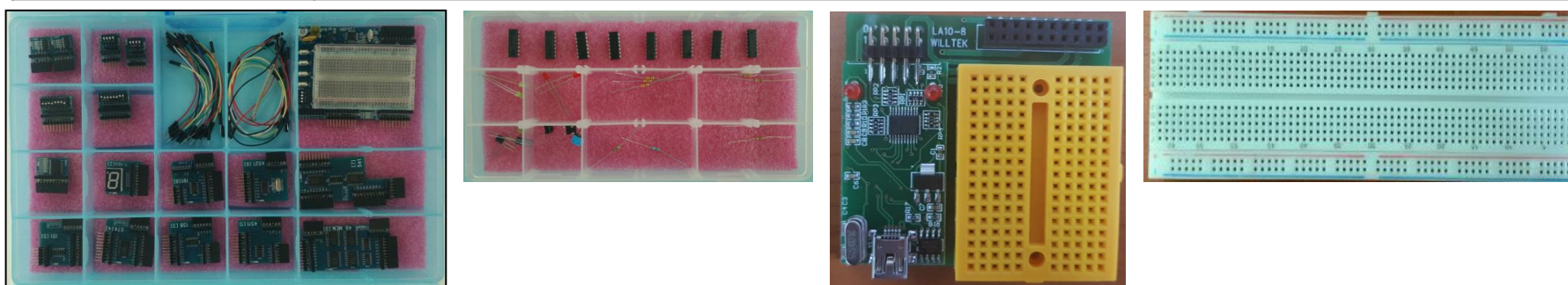


Part 6. Hardware 설계 분야 교육과정

Level 1

SW 개발자를 위한 디지털 회로 기초

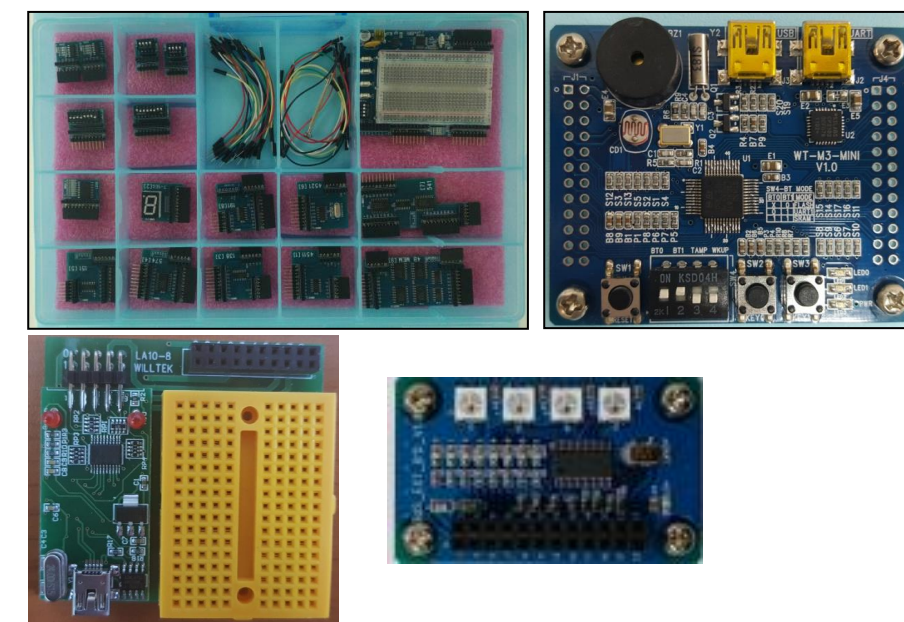
선수지식	없음	강의시간	40시간	실습장비	Logic Master Kit
과정개요	임베디드 시스템의 내부를 구성하는 기본적인 전자회로의 동작원리 및 프로세서와 메모리, 주변장치의 인터페이스를 위해 필요한 요소들에 대해 다루며, 가장 기초적인 소자의 동작부터 실제 프로세서와 주변장치를 제어하는 모든 과정에서 이론 및 실습을 병행하여 수행함으로써 교육생들이 하드웨어와 소프트웨어에 대한 융합지식을 습득하고 활용할 수 있도록 진행한다. 또한 진행되는 모든 내용에 대하여 동작원리를 배우고 직접 실습하여 체계적으로 지식을 심화하고, 실습장비가 실습별 모듈로 구성되어 원하는 주제만 실습도 가능하며, 필요시 해당 모듈들을 모두 연결한 전체 시스템 구성을 통해 학습한 전체적인 심화도 가능하도록 구성하였고, 특히 소프트웨어 지식만 가지고 있는 개발자도 디지털회로의 원리부터 심화응용까지 이해할 수 있도록 단계적으로 진행한다.				
교육목표	• PC기반이 아닌 임베디드 기반에서의 주변장치에 대한 이해 및 활용 능력 배양 • 제품에 사용되는 기초소자, 반도체소자, 범용디지털회로, 프로세서, 메모리, 주변장치의 동작원리 이해, 구성방법 점검 • 전원충격최소화, 노이즈제거, 오픈드레인, 풀업, 풀다운, 채터링, 슈미트트리거 등의 하드웨어 인터페이스 이슈에 대한 점검 • 스위칭, 타이밍분석, 디코더, 멀티플렉서, 버스, PLL, 메모리 I/F, 입출력확장 등의 이슈들에 대하여 분석 및 실습을 통한 점검 • 프로세서 기반 임베디드 시스템 개발에 필수적인 하드웨어 요소에 대한 점검과 실제 개발과정 사례를 통하여 실질적인 개발 이해				
교육내용	• 전기와 전원, 기본적인 전기소자의 연결과 동작원리, 다이오드와 LED, 전류와 전압의 분배 법칙, 트랜지스터의 동작원리, 스위칭 • 논리게이트와 디지털회로 분석 및 디지털 입출력 동작, 입출력회로: Push Pull과 Open Drain, 3 State, High Impedance • 플립플롭과 1bit메모리, 메모리셀, Clock, 타이밍차트 분석, DIP스위치, LED, Latch모듈의 구성, 카운터/타이머의 활용 • 채터링과 Debouncing회로, 슈미트트리거 입력, 574(Latch)활용 출력확장, 138(디코더) 활용 주소선택 • 7-segment 및 4511(BCD-to-7 segment)모듈 활용, Encoder, Multiplexer, Demultiplexer, 541(BUS driver) • 프로세서의 실행과 Clock, PLL, 일정범위 반복 카운터 및 전자주사위를 통한 동작분석 • 메모리와 메모리버스, CPU에의한 메모리 액세스, 임베디드 시스템과 임베디드 시스템의 제품화 과정				



Level 1

SW 개발자를 위한 임베디드 시스템 설계

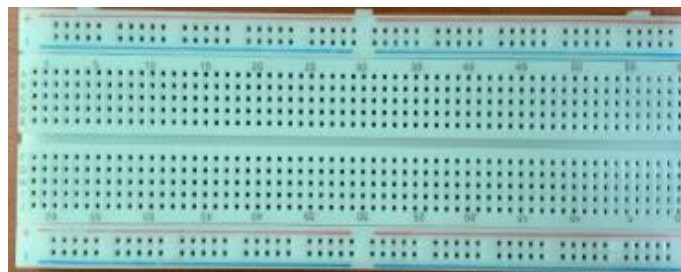
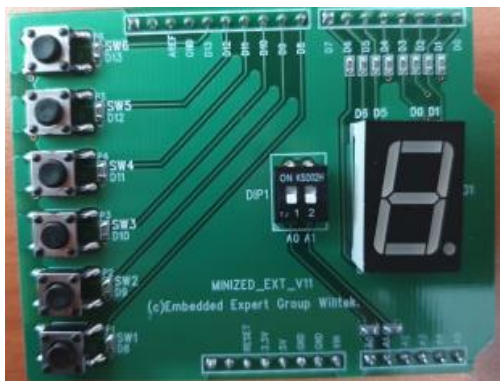
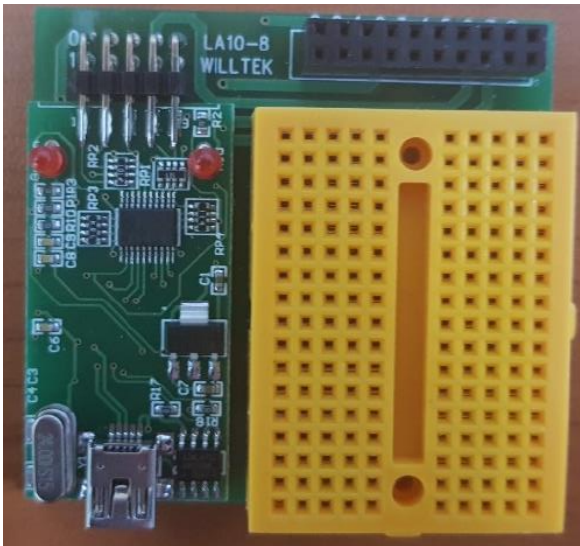
선수지식	없음	강의시간	40시간	실습장비	Logic Master Kit, Arm Cortex-M3 보드
과정개요	임베디드 시스템의 소프트웨어를 담당하는 개발자들은 하드웨어 지식이 필요 없을까? 예전에는 하드웨어와 소프트웨어 개발자들의 경계가 명확했던 편이었다. 그러나 임베디드 프로세서들이 등장하면서 PCB의 면적이나 원가 절감을 위하여 프로세서 외부에 연결하던 하드웨어(Glue Logic)나 주변장치들을 제조사들이 경쟁적으로 내부에 탑재하고 있다. 이러다 보니 소프트웨어 개발자들은 입출력 회로의 설정, 풀업, 풀다운 저항의 설정, 주파수 조절 등의 기능을 하드웨어가 아닌 프로그램으로 구현해야만 한다. 특히, 요즘은 하드웨어 설계자 없이 부품회사에서 제공하는 표준 회로도를 기반으로 제품을 설계하다 보니 회로에 대한 직접적인 이해 능력이 소프트웨어 개발자들에게 필수적으로 필요하게 되었다. 따라서 소프트웨어 개발자들이 높은 하드웨어 지식을 이해할 필요는 없지만 기본적인 하드웨어 설계에 관한 지식을 습득함으로써 원가절감 및 최적화된 성능 구현을 달성할 수가 있게 된다. 본 과정은 소프트웨어 개발자로서 반드시 알아야 하는 하드웨어 지식인 전원, 수동소자, 능동소자 동작 이해, 메모리 버스 인터페이스의 동작, 타이밍 차트 인식 등의 실무적인 하드웨어 지식을 습득할 수 있도록 한다. 특히, 실제 32비트 마이컴을 이용한 입출력 회로(LED 출력, KEY 입력), RS232, I2C, SPI 회로를 직접 Bread Board를 이용하여 구성해보고 프로그램으로 제어를 해보는 실습 등을 통하여 회로 인식 및 주변장치 제어 능력을 배양한다.				
교육목표	- 전원, 수동소자(RLC), 능동소자(다이오드, LED, TR, FET)의 역할을 이해하고 이를 인식할 수 있는 능력의 배양 - 입력 회로의 구성, 출력 회로의 구성을 이해하고 실제 프로그램에서 최적으로 설정할 수 있는 능력의 배양 - 기본적인 임베디드 시스템 회로 구성인 입출력, UART 통신, Clock 회로를 이해하고 실제 제어할 수 있는 능력 배양 - 필수 주변장치인 ADC, I2C 버스, SPI 버스 인터페이스의 동작 이해 및 이를 제어할 수 있는 능력 배양				
교육내용	- 전원, 저항, 콘덴서, 인덕터, 다이오드, LED, TR, FET의 역할 및 회로 인식 - Push Pull, Open Drain, 3-State 출력의 활용, Floating Input Pull Up, Pull Down 저항의 역할 - Chattering 현상, Schmitt Trigger Input, HW 및 SW에 의한 Chattering 방지 구현 - RS232 통신 규격 및 신호 규약, RS232 인터페이스 회로, USB-RS232 Bridge 회로의 이해와 활용 - Clock 생성 방법, PLL의 동작, Clock Divider의 역할 - 디지털 조합화로 및 순차회로의 이해와 CPU 메모리 인터페이스 타이밍 이해 - I2C 버스와 SPI 버스를 이용한 디바이스 확장 회로 이해 및 제어 프로그램 구현 - Timer의 동작과 구조 이해 및 PWM 출력 제어 프로그램 구현				



Level 1

Verilog를 이용한 디지털 회로 설계

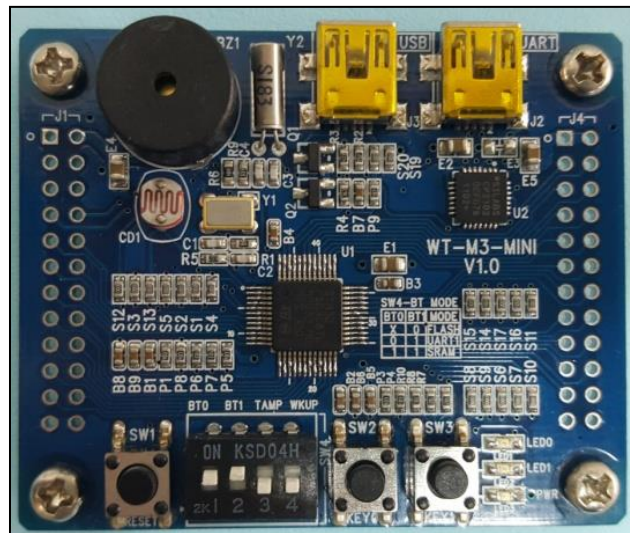
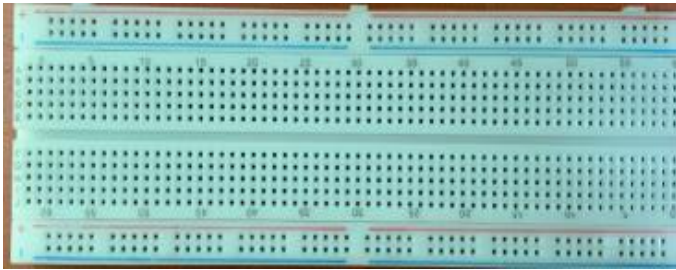
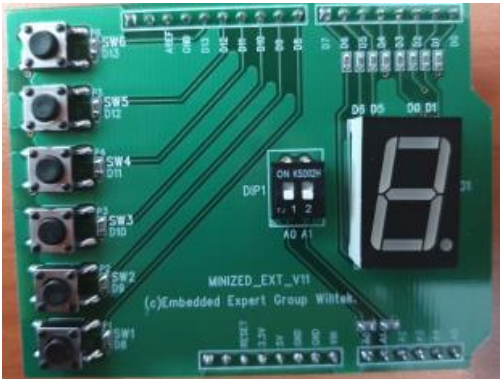
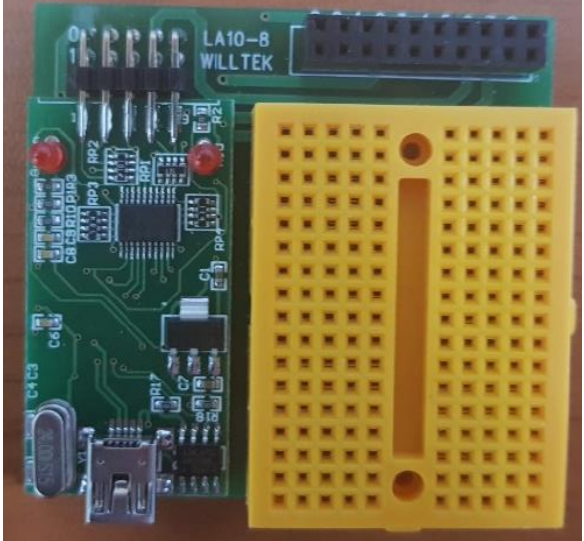
선수지식	디지털 회로 및 논리회로 이해	강의시간	40시간	실습장비	Mini-ZED 보드 Set
과정개요	본 과정에서는 기본적인 Verilog 디자인 및 테스트벤치 작성에 필요한 지식을 배울 수 있으며 Verilog 코드의 작성 및 문법 체크, 합성 및 시뮬레이션은 Xilinx FPGA 설계도구인 Vivado를 이용하여 수행하며 간단한 ALU를 내장한 Calculator를 설계하는 실습이 포함되어 있습니다. 강의 내용은 우선 Verilog를 구성하는 기본요소를 배우고 설계한 디자인이 기능적으로 올바른 동작을 하는지 검증하기 위한 코드인 테스트벤치의 개념과 작성방법을 학습합니다. 이어서 Verilog 에서 제공하는 각종 데이터 타입과 연산자 그리고 이들로 표현되는 연산식을 다루며 Verilog 구문의 대표적인 두 가지인 동시 대입문과 프로시저럴 블록과 하드웨어적인 동시 수행을 시뮬레이터가 모델링하는 방법 등에 대하여 학습합니다. 또한 Logic 알고리즘을 state diagram으로 표현하고 이로서 FMS 코드를 작성하는 방법에 대해서도 학습을 합니다.				
교육목표	• Verilog 문법을 이해하고 이를 이용한 회로 설계 능력을 배양한다 • 설계된 하드웨어 검증을 위한 테스트벤치의 개념과 작성 능력을 배양한다. • 실제 FPGA를 이용하여 설계된 하드웨어의 동작을 검증하는 능력을 배양한다.				
교육내용	• Verilog HDL Introduction, Hardware Modeling, Verilog 문법, Building Hierarchy, Operator/Expressions, • Data-Flow Modeling, Behavioral Modeling, Test Fixture, Test Bench 생성, Advanced Process Statements, RTL Verification • Finite State Machines, Targeting Xilinx FPGAs, Task and Functions, Verilog Post-Layout Timing Simulation • Mini Project				



Level 2

Verilog를 이용한 다양한 Peripheral 설계

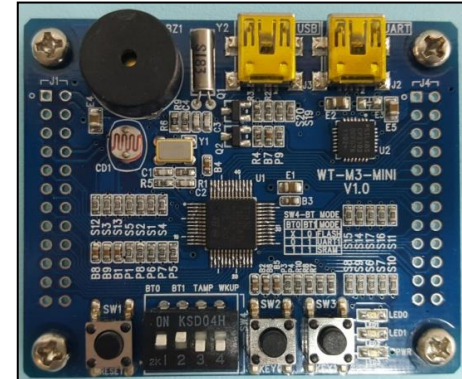
선수지식	Verilog 이해 및 기본 디지털 회로 설계 역량	강의시간	40시간	실습장비	Mini-ZED 보드 Set, Arm Cortex-M3 보드
과정개요	본 과정은 MCU 및 마이크로프로세서에 가장 필수적으로 탑재하는 Peripheral인 양방향 GPIO, UART 통신, Timer & Counter, SPI, I2C의 동작 및 구조를 학습하고 이러한 주변장치를 Verilog를 이용하여 직접 구현하고 시뮬레이션으로 검증한 후 실제 FPGA에 구현하여 동작을 확인하도록 하는 실무 위주의 교육 과정입니다. 이 교육 과정을 통하여 추후 SoC 개발을 위한 주변장치 설계 및 IP를 활용할 경우에 발생할 수 있는 타이밍 문제, 시뮬레이션과 실제 하드웨어 동작 간의 불일치 문제 등을 미리 실습을 통하여 경험할 수 있게 되어 현업 개발 업무에 많은 도움이 될 수 있게 됩니다.				
교육목표	• 양방향 GPIO, UART 통신, Timer & Counter, SPI, I2C의 동작 및 구조를 이해한다. • Verilog로 설계한 회로의 동작 테스트를 위한 시뮬레이션 및 실장 테스트를 수행하는 능력을 배양한다.				
교육내용	• GPIO 출력 장치와 GPIO 입력 장치를 설계하고 이를 FPGA에 구현한 후 마이컴과 연결하여 소프트웨어로 실제 동작을 시험 진행한다. • UART와 Timer 장치를 설계하고 이를 FPGA에 구현한 후 마이컴과 연결하여 소프트웨어로 실제 동작을 시험 진행한다. • SPI, S2C 버스의 동작을 이해하고 파형을 분석한 후 Device를 Verilog로 설계한 후 FPGA에 구현하여 마이컴 연동으로 동작을 시험한다.				



Part 7. Embedded System 분야 교육과정

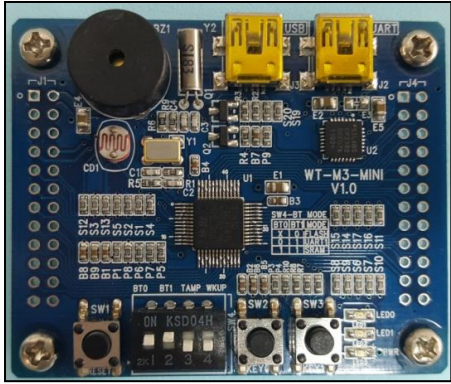
Level 1

임베디드 시스템 필수 디바이스 프로그래밍

선수지식	C기반 코드 구현 능력	강의시간	40시간	실습장비	Arm Cortex-M3 보드
과정개요	임베디드 프로그램은 PC와 같이 OS가 갖춰진 환경에서 개발하는 경우보다 전원 인가 후부터 응용 프로그램이 구동될 때까지 부트코드까지 직접 개발자가 개발하여야 한다. 리눅스와 같은 OS를 사용하는 경우라도 UBOOT와 같은 펌웨어는 직접 설계 혹은 수정해야 할 필요성이 있다. 또한, OS 없는 임베디드 환경에서 가장 기본적인 디바이스들을 제어하기 위하여는 주변장치 하드웨어의 구조에 대한 이해와 이를 효율적으로 제어하기 위한 능력의 확보가 필요하다. 본 과정은 최근 가장 많이 사용되고 있는 프로세서인 Cortex-M3 계열 프로세서를 이용한 실습보드를 활용하여 전원 인가 후부터 응용 프로그램이 구동될 때까지의 펌웨어를 직접 설계하고 가장 필수적인 디바이스인 범용 입출력장치, UART, Timer, 인터럽트, DMA를 직접 제어해 봄으로써 실무에서 어떠한 프로세서를 사용하게 되더라도 자신 있게 프로그램을 개발할 수 있는 능력을 배양한다. 특히, 본 과정을 통하여 그 동안 익힌 C언어가 어떻게 임베디드 시스템 제어에 사용되는지 그리고 디바이스 제어를 위한 하부함수의 필요성과 계층적 설계의 필요성을 느낄 수 있도록 하여 임베디드 기반 시스템 프로그래머로 성장할 수 있는 발판을 마련해 주게 된다. 특히, 단순 따라하기가 아닌 직접 프로세서 매뉴얼을 읽어보고 레지스터들의 기능을 분석하도록 하여 학습하지 않은 다른 프로세서를 활용하는 경우에도 프로그램 개발이 가능하도록 실무적으로 교육이 진행된다.				
교육목표	• 임베디드 시스템을 개발하기 위한 기초 작업인 개발 환경 구축, 컴파일러 설치 및 설정, JTAG 기반 장치의 활용 능력의 배양 • 프로세서의 전원 인가 후부터 응용 프로그램 구동까지 담당하는 부트코드의 역할 이해 및 부트 코드 설계 능력의 습득 • 필수 주변장치인 범용입출력 장치, UART, Timer 장치의 구조를 이해하고 이를 활용하는 디바이스 드라이버 설계 능력의 배양 • 시스템 성능 제고를 위한 인터럽트 및 DMA의 동작원리를 이해하고 이를 활용하는 응용 프로그램을 개발할 수 있는 능력의 배양 • C언어가 임베디드 시스템에 어떻게 적용되는가를 이해하여 임베디드 기반 프로그램 개발자로서의 C언어 활용 능력을 제고				
교육내용	• 범용 입출력(LED, Key 스위치) 제어 및 디바이스 드라이버 설계 • UART(RS232C) 제어 및 디바이스 드라이버 설계 • Timer & PWM 제어 및 디바이스 드라이버 설계 • 인터럽트 기반 프로그래밍(GPIO, UART, Timer) • DMA 기반 프로그래밍(SW 트리거, HW 트리거)  				

Level 2

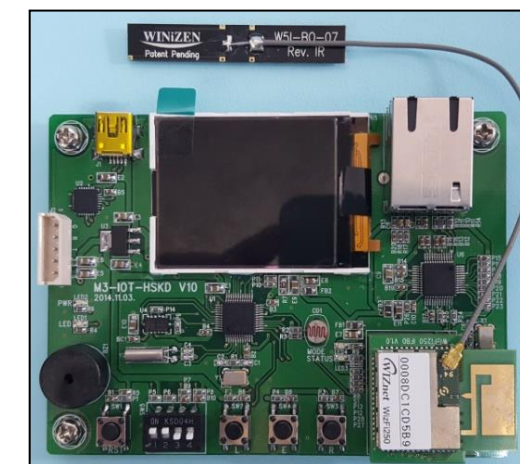
임베디드 시스템 확장 디바이스 프로그래밍

선수지식	필수 주변장치를 제어할 수 있는 지식	강의시간	40시간	실습장비	Arm Cortex-M3 보드
과정개요	임베디드 프로세서들은 매우 많은 주변장치들을 내장하고 있으며 이러한 주변장치들을 이용하여 다양한 기능을 구현하게 된다. 그러나 이러한 고급의 다양한 주변장치들을 잘 활용하기 위하여는 단지 소프트웨어 기술만이 아닌 매뉴얼의 분석 및 이해 그리고 제어를 위한 레지스터들의 이해 능력이 필요하다. 본 교육은 GPIO, UART, TIMER, 인터럽트, DMA와 같은 기본적인 필수 주변장치 이외에도 실무에서 많이 사용하는 주변장치인 ADC, PWM 출력을 비롯하여 I2C, SPI와 같은 범용 버스 인터페이스, 외부 메모리를 확장하기 위한 메모리 버스, SD Card 인터페이스 및 멀티미디어를 구현하기 위한 I2S, TFT LCD, Touch Panel 등을 활용하기 위한 주변장치들의 구조, 레지스터 이해, 디바이스 제어 프로그램 설계 방법 등에 대하여 교육한다. 특히, 본 교육은 실습에 사용된 임베디드 프로세서 이외의 여러 제조사에서 출시되는 다양한 제품의 주변장치 제어 능력을 배양하기 위하여 철저하게 프로세서 매뉴얼과 실습보드 회로도를 스스로 학습한 후 팀끼리 토론을 진행하면서 회로 결선부터 이의 제어 프로그램을 완성하는 워크샵 방식으로 진행된다.				
교육목표	- 임베디드 프로세서에 탑재된 다양한 주변장치들의 구조와 동작 원리를 이해할 수 있는 능력을 배양한다. - 주변장치의 내부 레지스터 동작을 이해하고 회로도를 이해하여 최적의 제어 프로그램을 개발할 수 있는 능력을 배양한다.				
교육내용	- 버스 : I2C, SPI 제어 - 타이머 : 타이머의 특수 기능(Capture, PWM 등) 활용 - 특수 메모리 : NAND Flash Memory, Serial EEPROM 인터페이스 회로 구현 및 제어 - 메모리 버스 확장 : SRAM 인터페이스 회로 구현 및 제어 - 멀티미디어 : SDI, I2S, TFT LCD, ADC & Touch Panel의 제어 - 그래픽 프로그램 구현을 위한 하부 Primitive 함수들의 구현 및 GUI 프로그램 설계 				

Level 2

이더넷과 WiFi를 이용한 IoT디바이스 설계 실무

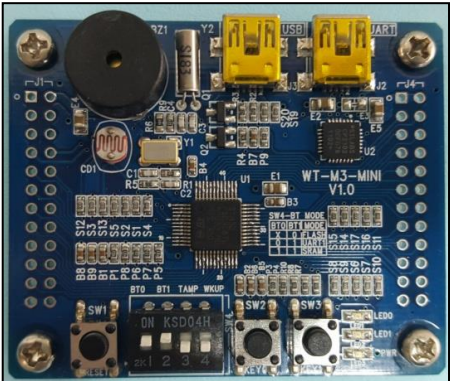
선수지식	C기반 코드 분석 및 구현 능력	강의시간	40시간	실습장비	Arm Cortex-M3 기반 IoT 보드
과정개요	본 과정은 스마트폰이나 리눅스가 탑재된 고가의 임베디드 시스템이 아닌 마이컴 기반의 저가 임베디드 시스템에서 IoT 디바이스를 개발하기 위해 필수적인 하드웨어 시스템, 소프트웨어, 네트워크 및 프로토콜에 대해 실무적으로 학습하는 과정이다. 이를 위하여 기본적인 임베디드 시스템에 필요한 사항들 및 센서 인터페이스 기술, 하드웨어 기반의 이더넷 및 WiFi를 이용한 인터넷 접속 기술 그리고 서버-클라이언트 통신 구현을 위한 IoT 기반 네트워크 프로토콜 구현 기술을 이론과 실습으로 교육한다. 최종적으로 IoT 디바이스의 센서 인터페이스를 통해 다양한 측정값을 수집하고, 해당 데이터들을 가공하여 최종적으로 연결된 유무선 네트워크를 통하여 전송하며, 특히 IoT 표준 프로토콜인 MQTT를 사용하여 신뢰성있는 서버와의 통신을 수행하는 기술들을 습득하고 활용할 수 있도록 한다.				
교육목표	- IoT 디바이스 개발을 위한 Cortex-M3 주변장치 제어 능력을 갖출 수 있다. - 프로세서의 다양한 버스 인터페이스 및 센서 인터페이스 기술을 습득할 수 있다. - 인터넷 프로토콜을 이해하고 H/W 이더넷 제어 기술을 습득할 수 있다. - H/W 기반 WiFi 모듈을 이용한 WiFi 인터페이스 기술을 습득할 수 있다. - IoT를 위한 경량 공개 네트워크 프로토콜(MQTT)을 활용하는 기술을 습득할 수 있다. - IoT 서비스 구축을 위한 서버와 디바이스 구조를 이해하고 활용할 수 있다.				
교육내용	- IoT개요 및 플랫폼 - 기본 디바이스 제어, 인터럽트 및 특수 디바이스 - 센서 인터페이스 - 통신과 네트워크, TCP/IP 프로토콜 - 소켓 프로그래밍, WiFi 모듈제어 - IoT 프로토콜 활용 - IoT 디바이스 구현실무				



Level 3

Arm Cortex-M 프로세서 시스템 프로그래밍

선수지식	Cortex-M 기반 프로그램 구현 경험	강의시간	40시간	실습장비	Arm Cortex-M3 보드
과정개요	Cortex-M 프로세서를 단순히 기존 마이컴 대체 목적으로만 사용할 것인가? 아니면 기존 마이컴에서 제공하지 못하는 안전하고 신뢰성 높고 프로그램을 개발할 목적으로 Cortex-M 프로세서를 사용할 것인가? 최근 자동차 및 산업용, 가전용 프로세서로 널리 사용되는 Cortex-M 계열 프로세서는 신뢰성 제고를 위한 특수한 기능들을 보유하고 있음에도 불구하고 기존 마이컴보다 성능 좋고 사용하기 쉬운 마이컴 정도로만 인식되어 훌륭한 성능을 제대로 활용하지 못하고 사용하는 실정이다. 본 과정은 제품의 신뢰성 제고와 효율 향상을 위하여 Cortex-M 계열 프로세서 기반 소프트웨어 개발자들이 알아야 할 Cortex-M 프로세서에 특화된 다양한 고급 디버깅 기술 및 시스템 안전과 신뢰성 제고를 위한 고급 시스템 프로그래밍 기술을 습득할 수 있도록 하며 특히 고도의 신뢰성을 요하는 자동차, 산업용, 군사용, 우주항공용 제품 개발자들은 반드시 마스터 해야 하는 필수 교육 과정이다.				
교육목표	• 임베디드 Cortex-M 프로세서 기반 최적화된 시스템 프로그램 설계를 위한 능력을 확보할 수 있다. • 시스템 오류(Fault)가 발행하는 원인을 이해하며 Fault 발생 위치와 원인을 추적할 수 있다. • 작성한 코드가 정상 동작하지 않을 경우 어셈블리어 레벨에서 고급 디버깅을 할 수 있다. • 인터럽트 기반 프로그램 설계를 위하여 인터럽트와 중첩 인터럽트 핸들러를 설계할 수 있다. • 배터리 전원을 이용하는 시스템을 위하여 ARM에서 제공하는 저전력 제어 모드를 활용할 수 있다.				
교육내용	• Cortex-M 계열 프로세서의 특징점, 표준 메모리 맵, 부트 코드, 메모리 레이아웃, CRT Startup • v7M 어셈블리 명령어, CMSIS 규정, Bit Banding, IAP 부트코드 • AAPCS 규정, C-ASM 상호 호출, C측 변수 공유, 동작 모드, Thread와 Handler • Fault(Memory Management Fault, Bus Fault, Usage Fault, Hard Fault) 발생 이유 • Fault 발생 원인 추적을 위한 핸들러 설계, System Lockup, OS가 사용하는 Exception(NMI, Pendsv, SVCall, SysTick) • 인터럽트와 중첩인터럽트, 우선순위 정책, 저전력 제어 모드				



Level 4

Arm Cortex-A 멀티코어 시스템 플랫폼 설계

선수지식	Cortex-A 기반 프로그램 개발 경험	강의시간	40시간	실습장비	Cortex-A Quad Core 보드
과정개요	최근 임베디드 시스템을 탑재하는 시스템들은 Cortex-A(v7) Multicore 프로세서를 많이 활용하고 있는 상황이다. 그러나 많은 경우 Cortex-A Multicore 프로세서의 구조를 정확히 이해하지 않고 포팅된 리눅스 환경에서 응용 프로그램이나 디바이스 드라이버를 개발하다 보니 부트코드 분석, 고급 디버깅 및 최적화된 코드 설계시에 많은 어려움을 느끼게 된다. 그러나 막상 이에 대한 학습을 하려 해도 교재나 인터넷 자료 등도 잘 오픈 되어 있지 않으며 SoC 제조사 및 ARM사에서 제공되는 매뉴얼만으로는 구조의 이해가 쉽지 않은 것이 현실이다. 본 과정은 이러한 문제를 극복하기 위하여 Cortex-A Single Core에 대한 구조, 동작, AAPCS 규정을 먼저 이해한 후 Multicore를 위한 부트코드, Exception 및 인터럽트 처리, Snoop Controller를 포함한 코프로세서의 동작 등을 학습하여 OS 커널 분석, 부트코드 분석 및 설계, 고급 디버깅 등에 활용할 수 있도록 한다.				
교육목표	• Cortex-A(v7) Multicore 기반 프로세서의 BIOS부터 응용프로그램이 실행되기까지의 전 과정을 이해할 수 있다. • AAPCS 규칙에 의거한 완벽하게 C와 호환되는 최적화된 어셈블리 기반 함수를 설계할 수 있다. • 시스템 오류가 발생할 경우 원인 추적을 위한 핸들러를 설계할 수 있다. • ARM의 Exception을 이해하고 이의 처리 핸들러를 설계할 수 있다. • 멀티프로세싱 OS에서 반드시 필요한 코프로세서(MMU, Cache)의 동작을 이해하고 제어 프로그램을 설계할 수 있다. • Cortex-A 기반 시스템의 오류를 추적하기 위한 고급 디버깅 기술을 확보할 수 있다. • 멀티프로세서의 동기화 및 멀티부팅, 불일치 해소 등을 위한 기법을 이해하고 이를 활용할 수 있다.				
교육내용	• Cortex-A 멀티코어 프로세서, ARM 어셈블리어, 멀티코어 부트코드 설계 • AAPCS 규정, ARM 모드와 Exception, Exception 핸들러 설계와 원인 분석 • 시스템 콜, OS와 Exception, 인터럽트 핸들러 • CP15와 코프로세서, MMU의 3가지 역할, Cache의 동작과 Cache 정책 수립, 코프로세서의 제어와 활용 • 멀티코어의 SCU의 구조와 제어, GIC의 구조와 제어, 멀티코어의 분산처리 활용				



Level 2

FreeRTOS 커널 포팅 및 멀티태스킹 응용 프로그램 개발

선수지식	C기반 코드 구현 능력	강의시간	40시간	실습장비	Arm Cortex-M7 ULTRA 보드
과정개요	IoT장치, 자동차 전장 제품, 스마트 가전기기, 산업용 기기 등 정해진 시간내에 입력을 처리해야 하는 실시간 시스템들을 위한 SoC 출시가 늘어나고 이를 체계적으로 지원하기 위하여 RTOS를 기반으로 하는 플랫폼의 개발 및 지원이 확대되어 지고 있으며, 이러한 플랫폼들은 다양한 제품 개발을 지원하고 다른 플랫폼과의 차별화를 위해 기본적인 네트워크 프로토콜 및 파일시스템 지원을 강화하고 있다. 본 과정은 기본적인 OS/RTOS에 대한 지식을 갖추고 있는 개발자를 대상으로 현재 가장 널리 사용되는 무료 RTOS중 하나인 FreeRTOS를 ARM사의 Cortex-M3 프로세서가 탑재된 타겟 시스템에 포팅 하기 위한 절차와 응용프로그램 개발을 위한 다양한 내용들에 대해 다루며, 특히 표준 코드를 시작으로 RTOS의 포팅을 단계별로 진행하면서 발생할 수 있는 이슈들에 대해 점검하고, TCP/IP 네트워크 및 FAT16 기반의 파일시스템을 실습보드에 적용해봄으로써 실제 FreeRTOS 기반 제품개발 시 겪을 수 있는 주요사항들을 점검해본다.				
교육목표	• 기본적인 실시간 운영체제 기반의 플랫폼과 RTOS의 커널 구조에 대해 이해할 수 있다. • 커널의 기본구조 및 태스크간 통신을 위한 ITC 등 이벤트를 처리할 수 있는 능력을 갖출 수 있다. • FreeRTOS를 임베디드 시스템에 포팅하고 멀티태스킹 기반의 응용 프로그램을 설계하는 기본 능력을 갖출 수 있다. • TCP/IP 프로토콜의 기본구조 및 소켓프로그래밍을 활용한 네트워크 통신을 구현할 수 있다. • FAT16 파일시스템을 이해하고 실제 구현할 수 있다. • 네트워크와 파일시스템을 활용한 멀티태스킹 기반 응용프로그램을 구현할 수 있다.				
교육내용	• RTOS 개요, 개발환경 구축, 실시간 시스템의 이해 • Cortex-M3 프로세서, Cortex-M3 어셈블리어 요약, Mode와 Exception, 인터럽트와 시스템 콜 • FreeRTOS 커널 포팅, FreeRTOS 커널 분석 • 태스크 관리, 이벤트 처리 구조, 태스크간 통신(ITC) • 멀티태스킹, 응용 프로그램 구현				



Level 4

OS 만들면서 배우는 Cortex-A 코프로세서 구조 및 제어

선수지식	Cortex-A 기반 프로그램 개발 경험	강의시간	40시간	실습장비	Cortex-A Quad Core 보드
과정개요	최근 모바일 및 디지털 가전, 스마트 기기 등에는 Cortex-A 프로세서에 리눅스를 탑재하여 개발되는 추세이다. 본 과정은 ARM 어셈블리어, Exception 핸들러, AAPCS 규정에 의한 C와 호환되는 어셈블리 함수 설계, 코프로세서(MMU, Cache) 제어 프로그램 구현 기법을 학습한다. 학습한 내용을 기반으로 리눅스 및 멀티프로세싱 OS 커널에 사용되는 보편적인 모듈인 Demand Stack, Demand Paging, Multi Processing Scheduler, Resource Manager의 기본적인 동작을 Cortex-A 프로세서에 직접 구현해 봄으로써 OS 커널의 이해 및 분석 시에 도움이 되도록 하며 오류 발생 원인을 추적하기 위한 고급 디버깅 스킬을 학습한다.				
교육목표	- ARM 프로세서가 탑재된 임베디드 시스템의 오류 발생시 이를 분석할 수 있는 고급 디버깅을 할 수 있다. - ARM 프로세서의 Exception 발생 원인 분석과 Exception 핸들러를 설계할 수 있다. - MMU와 Cache의 구조와 동작을 이해하고 이를 제어하는 프로그램을 설계할 수 있다. - OS 커널에 사용되는 Demand Stack, Demand Paging, Multi Processing, Scheduler, Resource Manager를 분석할 수 있다. - OS 커널에서 사용하는 이벤트 핸들러, 디바이스 드라이버 시스템 콜 핸들러의 구조와 동작을 이해할 수 있다. - 리눅스 및 기타 OS 포팅 및 커널 분석, 디바이스 드라이버 설계 시 필요한 고급 지식을 습득할 수 있다.				
교육내용	- ARM 어셈블리 명령어 - C-ASM 인터페이스 - MMU 제어 - Cache와 WB 제어 - Resource Management - Memory Management - OS 커널 핵심 요소들의 설계 				

Part 8. Embedded Linux 분야 교육 과정

Level 1

재미있게 배우는 임베디드 리눅스 활용

선수지식	C 문법 이해, C 코드 구현	강의시간	40시간	실습장비	Cortex-A Quad Core 보드
과정개요	리눅스는 임베디드 분야에서 기술을 습득하는데 많은 시간이 걸리고, 특히 처음 접하는 사용자는 너무 어렵게 느껴지기 때문에 초기에 포기하는 경우가 많이 가장 많이 쓰이는 운영체제이므로 개발자들이 필수적으로 익혀야 할 기술이 되었다. 그러나 리눅스는 다루는 분야가 매우 넓기 때문에 발생하고 있다. 하지만 이는 리눅스가 어렵기 때문이 아니라 생소하기 때문인 경우가 대부분이어서, 오히려 리눅스 환경에 어느 정도 익숙해 진 후에는 리눅스의 강력한 기능에 감탄하게 되고 좀더 공부하고 싶은 의욕이 생기게 된다. 따라서 리눅스 초보자들에게는 세부적이고 전문적인 내용을 공부하기 전에 리눅스 전반적인 기본 지식의 습득함으로써 리눅스에 익숙해질 필요가 있다. 본 과정에서는 리눅스 환경에 적응할 수 있도록 임베디드 리눅스 환경, 응용 프로그램, 커널, 디바이스 드라이버, 멀티미디어 등 전반적인 내용을 다루되 깊이 파고들기 보다는 폭넓은 내용을 이해하는 방향으로 수업이 진행된다. 본 과정을 통해 리눅스에 익숙해지게 되고 추후 세부 분야로 들어갈 수 있는 기틀을 마련할 수 있다. 또한 임베디드 실습 보드를 사용함으로써 임베디드 환경을 이해하고 활용하는 능력을 향상시킬 수 있고, 쉬운 예제 중심으로 실습이 이루어지므로 리눅스 경험이 전혀 없는 초보자도 쉽게 따라올 수 있다.				
교육목표	• 리눅스 명령어, 셸 스크립트, 개발 환경 구축 등 임베디드 리눅스의 사용에 필요한 기초 지식을 습득한다. • 프로세스, 시그널, 프로세스 간 통신, 스레드 등 리눅스의 특성을 이해하고, 이들을 활용한 효율적인 프로그램을 작성할 수 있다. • 리눅스 커널의 특성을 파악하고, 디바이스 드라이버의 구조와 동작 원리를 이해할 수 있다. • 리눅스에서 제공하는 멀티미디어 기능을 임베디드 환경에서 활용하는 방법을 익힌다.				
교육내용	• 임베디드 리눅스 환경의 이해: 리눅스 명령어, 셸 스크립트, 개발 환경 구축 • 리눅스 시스템 프로그래밍: 프로세스, 시그널, 세마포어, 공유 메모리, 메시지 큐, 스레드 • 리눅스 커널 개요: 프로세스 관리, 시스템 콜, 메모리 관리, 파일시스템 • 디바이스 제어: 문자 디바이스, 하드웨어 제어, 인터럽트 • 디바이스 모델, 블록 드라이버, 네트워크 드라이버 개요 • 리눅스 멀티미디어: 사운드, 디스플레이, Video4Linux, 입력 장치, Qt				



Level 1

실전 임베디드 리눅스 Application 프로그래밍

선수지식	C기반 코드 구현 능력	강의시간	40시간	실습장비	Cortex-A Quad Core 보드
과정개요	임베디드 리눅스 기반의 제품을 개발할 때 응용 프로그램이 차지하는 비중은 매우 크다. 응용 프로그램이 얼마나 효율적이고 안정적이나에 따라 제품의 성패가 결정된다고 해도 과언이 아니다. 그럼에도 불구하고 많은 개발자들은 임베디드 기반의 제한적인 환경과 동작 원리를 완벽하게 파악하지 못한 상태에서 응용 프로그램을 작성함으로써, 비록 기능 구현이 되었다 하더라도 효율적이지 못하거나 문제점을 내포한 결과물을 얻을 가능성이 크다. 또한 상용의 응용 프로그램이나 남이 작성한 응용 프로그램을 내가 작성한 응용 프로그램과 결합해야 하는데 응용 프로그램들을 운영하고 제어하는 방법을 모른다면 어려움을 겪을 수 밖에 없고, 작성한 응용 프로그램에 문제가 발생한 경우 원인 파악이 필요한데 응용 프로그램과 리눅스 커널과의 관계를 제대로 이해하지 못한 상태에서는 디버깅이 쉽지 않다. 본 과정에서는 프로세스와 스레드의 관리 기법, 이들간의 데이터 교환과 동기화 방법 등 리눅스가 제공하는 특징과 장점을 활용하여 효율적인 응용 프로그램을 작성하는 방법을 익힐 수 있고, 여러 개의 응용 프로그램을 결합하여 효율적으로 운용하고 제어하는 기법을 습득할 수 있으며, 리눅스 커널을 중심으로 한 전체 시스템 레벨에서 응용 프로그램을 분석하는 능력을 향상시킬 수 있다. 또한 임베디드 실습 보드를 사용함으로써 임베디드 환경의 이해도와 활용도를 높일 수 있고, 예제 중심의 실습으로 난해한 내용을 쉽게 이해할 수 있으며, 다양한 연습 문제들을 구현해 봄으로써 배운 내용을 적용하고 활용하는 실무 능력을 향상시킨다.				
교육목표	- 리눅스 명령어와 셸 스크립트를 이해하고 활용할 수 있다. - 임베디드 리눅스 개발 환경을 구축해 봄으로써 리눅스 시스템의 동작 순서와 원리를 이해한다. - 임베디드 리눅스 환경에서 프로그램을 개발하고 사용하는 방법을 습득한다. - 프로세스 관리 및 시그널 활용을 통해 효율적인 응용 프로그램을 작성할 수 있다. - 프로세스 간 통신, 스레드 활용 및 동기화 방법 등 실무에 꼭 필요한 기법을 습득한다.				
교육내용	- 리눅스 명령어, 셸 스크립트 - 임베디드 리눅스 개발 환경 구축 및 응용 프로그램 개발 방법 - 프로세스, 시그널 - 파이프, 세마포어, 공유 메모리, 메시지 큐 - 스레드				



Level 1

리눅스 명령어 및 셸 프로그래밍

선수지식	없음	강의시간	40시간	실습장비	없음
과정개요	본 과정은 리눅스 시스템을 사용하는 개발자와 관리자를 위한 과정이다. 리눅스 시스템에서는 윈도우와 달리 터미널을 이용하여 셸에 직접 명령을 입력하여 작업을 하는 것이 필요하며, 이를 위해 명령의 사용 법 및 좀 더 유용한 형태로 명령을 실행하기 위해 셸 프로그래밍을 사용할 수 있다. 셸 프로그래밍은 자주 사용하는 명령을 프로그램으로 작성하고 이를 필요할 경우에 실행하는 것입니다. 리눅스 시스템에서 셸 스크립트는 매우 유용한 도구입니다. 하지만 셸 스크립트가 많이 어렵지 않음에도 불구하고 문법을 모르기 때문에 사용하기 어려운 경우가 종종 발생하게 됩니다. 셸 스크립트를 잘 사용하게 되면 리눅스 시스템에서 필요한 기능을 프로그래밍 없이 구현할 수 있으므로 셸 스크립트 사용 능력은 리눅스 시스템 관리자나 개발자에게 필수적인 요소이다. 본 과정을 통해 명령의 사용법 및 셸 스크립트를 작성하고 해석하고 활용하는 능력을 확보할 수 있습니다.				
교육목표	• 리눅스 기본 명령어와 확장 명령어를 이해하고 사용할 수 있다. • 셸 스크립트의 기본 문법과 확장 문법을 습득하여 다양한 셸 스크립트를 작성할 수 있다. • 다양한 예제 구현을 통해 실무에 적용 가능한 셸 스크립트 작성을 원할하게 할 수 있다. • 리눅스 시스템 관리에 필수적인 셸 스크립트를 분석할 수 있고 환경에 맞게 수정할 수 있다.				
교육내용	• 리눅스 기본 명령어 • 리눅스 확장 명령어 • vi 에디터 • 셸 스크립트 문법 • 셸 스크립트 연습 및 활용				

Level 2

리눅스 IPC의 이해와 응용

선수지식	리눅스 기본 활용 능력	강의시간	40시간	실습장비	없음
과정개요	리눅스 환경에서 응용프로그램을 개발하거나 시스템 프로그램을 작성할 때 다양한 IPC를 상황에 맞게 적용하는 것이 매우 중요하다. 특히, 프로세스 사이의 동기화 및 동시성 문제가 발생할 경우 IPC 동작의 정확한 원리를 이해하지 못한 경우에는 문제를 해결하기가 힘든 실정이다. 본 과정에서 다양한 IPC를 실제로 실습함으로써 각 IPC의 사용법과 장단점을 파악하여 실무에 활용할 수 있도록 한다				
교육목표	• 리눅스 환경에서 응용 프로그램을 개발하거나 프로세스 간 필요한 작업을 수행할 때 필수적인 IPC의 개념을 습득한다. • 프로세스, 시그널, 파이프, IPC, 스레드 등 리눅스의 핵심 요소들을 활용한 프로그래밍을 구현 할 수 있다. • 프로세스 관리 및 제어를 위한 다양한 IPC의 통신원리, 사용방법 및 동기화 등 실무에 꼭 필요한 기법을 습득한다. • 실제 각각의 IPC를 사용한 실습예제를 분석하고 구현함으로써 리눅스 시스템에서의 활용능력을 배양한다.				
교육내용	• 쉘 스크립트 기본 • 응용 프로그램 • 프로세스, 시그널 • 파이프, 세마포어, 공유 메모리, 메시지 큐 • 스레드 • 소켓				

Level 2

임베디드 리눅스 포팅 및 디바이스 드라이버

선수지식	C 코드 구현, vi 에디터, 리눅스 명령어	강의시간	40시간	실습장비	Cortex-A Quad Core 보드
과정개요	리눅스 포팅은 기존 보드에서 동작하는 리눅스를 신규 보드에서 동작하도록 이식하는 작업이다. 리눅스 포팅 이후에는 기존 주변 장치의 디바이스 드라이버 수정과 신규 주변 장치의 디바이스 드라이버 개발이 필수적으로 뒤따르게 된다. 과거에는 기존 보드와 신규 보드의 CPU가 달라 포팅 작업이 어려웠지만 최근 들어 CPU 제조사의 지원이 빨라지면서 CPU의 변경 없이 주변 장치만 변경하면 되므로 포팅 작업이 다소 수월해졌다. 그러나 포팅 작업은 하드웨어와 연관되어 있고 부트로더와 커널의 동작 원리를 이해해야 하므로 여전히 쉽지 않은 분야임이 확실하다. 또한 주변 장치가 변경되면 디바이스 드라이버가 신규로 작성되거나 포팅되어야 하는데 이는 디바이스 드라이버의 동작 원리를 이해할 수 있고 하드웨어 제어 기술을 구비한 상태에서 가능하므로 혼자서 접근하기에는 시간이 많이 필요하게 된다. 이러한 기술은 교육을 통해 습득할 수 있으며 이를 통해 임베디드 리눅스 기반 제품의 개발 기간을 줄이고 제품 원가를 낮출 필요가 있다. 본 과정에서는 임베디드 기반 보드에 리눅스를 포팅하는 기술과 디바이스 드라이버를 작성, 분석, 포팅하는 기술을 다룬다. 리눅스 포팅을 위해 필수적인 부트로더 분석, 리눅스 커널 초기화 과정 이해, 루트 파일시스템 구축 등을 진행하며 이러한 지식을 바탕으로 시스템을 디버깅하는 능력을 향상시킬 수 있다. 또한 디바이스 드라이버의 동작 원리를 이해함으로써 신규 디바이스 드라이버를 작성할 수 있고, 제조사에서 제공하는 디바이스 드라이버를 포팅할 수 있으며, 디버깅이 가능할 정도로 디바이스 드라이버 분석 능력을 향상시킬 수 있다.				
교육목표	• 부트로더의 역할과 동작을 이해하고 부트로더를 활용하여 시스템을 구축할 수 있다. • 리눅스 커널의 부팅 과정을 이해하고, 타겟 시스템에서 동작하도록 리눅스 커널을 포팅할 수 있다. • 디바이스 드라이버의 동작 원리를 이해하고, 디바이스 모델의 구조에 맞게 디바이스 드라이버를 작성할 수 있다. • 블록 드라이버와 네트워크 드라이버를 이해하고 분석할 수 있으며, 이더넷 드라이버 포팅을 통해 포팅 기술을 습득한다.				
교육내용	• 리눅스 커널 개요: 프로세스 관리, 시스템 콜, 메모리 관리, 파일시스템 • U-Boot의 이해와 활용 • 리눅스 커널 포팅: 커널 초기화 과정, 머신 추가, 루트 파일시스템 구축 • 디바이스 제어, 하드웨어 인터페이스와 인터럽트 • 디바이스 모델, 플랫폼 드라이버 • 블록 드라이버, 네트워크 드라이버				



Level 2

임베디드 리눅스 멀티미디어 & 네트워크

선수지식	C 코드 구현, vi 에디터, 리눅스 명령어	강의시간	40시간	실습장비	Cortex-A Quad Core 보드
과정개요	수많은 멀티미디어와 네트워크 제품들이 임베디드 리눅스를 탑재하고 있다. 그 이유는 리눅스가 자체적으로 뛰어난 멀티미디어와 네트워크 기능을 보유하고 있고, 자유롭게 사용 가능한 상용 프로그램들을 제공하고 있기 때문이다. 따라서 리눅스 개발자들은 상용의 멀티미디어와 네트워크 프로그램들을 활용하는 능력을 갖추는 필요가 있다. 그러나 임베디드 시스템에서는 상용 프로그램을 활용하는 능력 이외에도 이를 임베디드 환경에 맞도록 포팅 및 최적화하는 기술을 갖추어야 하고, 신규로 프로그램을 작성할 수 있어야 하며, 경우에 따라 디버깅이 가능한 수준까지 기술을 확대시켜야 한다. 또한 멀티미디어와 네트워크 기능이 커널 내부와 디바이스 드라이버에서 어떻게 처리되는지 흐름을 이해할 필요가 있다. 특히 네트워크는 환경적인 요인에 의해 문제가 많이 발생하므로 프로그램을 작성하는 시점부터 원칙을 지켜야 하고 항상 예외 상황을 염두에 두어야 한다. 본 과정에서는 임베디드 실습 보드 기반으로 상용의 멀티미디어와 네트워크 프로그램을 포팅하고 활용하는 방법을 익히고, 응용 프로그램과 커널 및 디바이스 드라이버 간 연관 관계를 이해하며, 데이터의 흐름을 분석하여 디버깅이 가능한 수준까지 기술을 향상시킬 수 있다. 특히 멀티미디어 환경에서 다양한 응용 프로그램들을 통합하고 제어하는 방법을 익히고, 네트워크를 이용하여 원격에서 장치를 제어해 봄으로써 임베디드 환경에서 네트워크가 어떻게 사용되는지 이해하고 이를 실무에 활용할 수 있다.				
교육목표	- 리눅스에서 제공하는 멀티미디어 기능을 타깃 시스템에서 사용할 수 있다. - Qt 프로그램 작성법을 익히고, Qt에서 멀티미디어를 구현하는 기술을 습득한다. - TCP/UDP 기반의 소켓 프로그램을 반이중, 전이중, 다중 클라이언트 등 다양한 환경에서 작성할 수 있다. - 네트워크 응용 프로그램 간 주고 받는 통신 프로토콜을 설계하여 원격으로 임베디드 장치의 디바이스를 제어할 수 있다.				
교육내용	- 사운드 시스템, 디스플레이 - Video4Linux, 입력 장치 - Qt 프로그래밍 - TCP/IP 프로토콜, 소켓 프로그래밍 - 디바이스 드라이버 개요, 원격 디바이스 제어 - 표준 네트워크 서비스 구축 및 활용				



Level 2

임베디드 시스템 및 리눅스 파일 시스템의 이해

선수지식	리눅스 기본 명령이해 및 기본 C 코드 설계	강의시간	40시간	실습장비	Cortex-A Quad Core 보드
과정개요	본 과정은 임베디드 시스템 및 리눅스 시스템에서 사용되는 파일 시스템을 직접 구축하고 파일 시스템의 구조와 동작을 이해하는 과정이다. 본 과정에서는 PC에서 범용적으로 사용되는 FAT16, FAT32 파일시스템의 구조를 이해하고 SD카드 기반 FAT16, FAT32를 직접 코드로 구현해봄으로써 파일시스템의 구조와 동작원리 등을 이해할 수 있다. 또한 임베디드 리눅스 시스템의 파일 시스템을 이해하기 위하여 Cortex-A9 Quad Core 기반의 실습보드를 활용하여 직접 SD 카드에 EXT4 파일 시스템을 구축하고 리눅스 명령으로 파일 시스템을 관리하며 메모리 덤프 실습을 통하여 EXT4 파일 시스템의 구조를 직접 분석한다. 또한 리눅스 커널의 VFS 계층을 이해하고 커널의 파일 시스템 코드를 분석함으로써 파일 시스템에 대한 구조를 이해할 수 있고 동작 원리를 파악할 수 있다.				
교육목표	• 범용적으로 사용되는 FAT16, FAT32 파일시스템의 구조와 동작 원리를 이해한다 • 임베디드 리눅스에서 사용되는 VFS와 EXT4 파일 시스템의 구조와 동작원리를 이해한다 • 리눅스 커널 레벨에서 EXT4 파일시스템을 디버깅할 수 있는 능력을 배양한다				
교육내용	• FAT16, FAT32의 구조 이해 • FAT16, FAT32의 코드 설계 • 임베디드 리눅스 개발 환경의 이해 및 구축 • 리눅스 파일시스템 개요 • 리눅스 명령을 통한 파일시스템 관리 • EXT4 파일시스템 구조 이해 및 분석 • 리눅스 커널의 VFS 계층 이해 • 리눅스 커널의 파일시스템 코드 분석 • 리눅스 커널 레벨에서 EXT4 파일시스템 디버깅				



Level 3

고급 임베디드 리눅스 디바이스 드라이버

선수지식	리눅스 시스템 이해 및 드라이버 동작 이해	강의시간	40시간	실습장비	Cortex-A Quad Core 보드
과정개요	임베디드 리눅스를 이용하여 제품을 개발하면 기능 구현을 쉽고 빠르게 할 수 있다고 생각하지만 실상은 그렇지 않다. 특히 디바이스 드라이버의 경우 기존 소스를 검증 없이 그대로 사용하는 경우가 많은데 이러한 경우 개발 중 또는 양산 후 발생하는 버그에 대처하기 힘들게 된다. 또한 대부분의 제품에는 신규 디바이스가 사용되므로 신규 디바이스 드라이버의 원천적인 개발 능력이 필요하고, 제조사에서 제공하는 디바이스 드라이버의 포팅 및 최적화 기술도 요구된다. 따라서 디바이스 드라이버 개발 능력을 갖추어 이러한 문제를 해결하고 나아가 보드 설계 시 디바이스를 합리적으로 선정함으로써 제품 원가를 줄이고 제품 성능을 향상시킬 필요가 있다. 본 과정에서는 임베디드 실습보드를 통하여 디바이스 드라이버를 직접 작성하고, 분석하고, 포팅하는 능력을 향상시키는데 중점을 두고 있다. 즉, 디바이스 드라이버의 동작 원리를 이해한 후 원리에 입각하여 디바이스 드라이버를 작성하는 방법을 습득하고, 제조사에서 제공하는 디바이스 드라이버를 현재 시스템에 포팅하는 기술을 익히며, 디바이스 드라이버의 구조와 실행 순서를 파악하여 디버깅 능력을 향상시키고, 디바이스 드라이버 작성 시 유용한 기법과 필수적인 요소 등 실무에 바로 적용할 수 있는 기술을 확보할 수 있다.				
교육목표	• 디바이스 드라이버의 구조와 동작 원리를 이해한 후, 디바이스 드라이버의 작성 방법을 이해한다. • 시간 관리, 지연된 작업, 커널 동기화 등 디바이스 드라이버 작성 시 필수적인 요소들을 이해하고 활용할 수 있다. • 디바이스 모델의 구조를 이해하고 그 형식에 맞게 플랫폼 드라이버를 작성할 수 있다. • 블록 드라이버와 네트워크 드라이버를 이해하고 분석할 수 있으며, 이더넷 드라이버 포팅 실습을 통해 포팅 기술을 습득한다. • 리눅스 멀티미디어 기능과 디바이스 드라이버와의 관계를 이해한다.				
교육내용	• 리눅스 커널 개요: 프로세스 관리, 시스템 콜, 메모리 관리, 파일시스템 • 디바이스 드라이버 개요, 문자 드라이버의 이해와 구현 • 하드웨어 인터페이스와 인터럽트 • 커널 자원 활용: 시간 처리, 지연된 작업, 커널 동기화, 메모리 매핑 • 디바이스 모델, 플랫폼 드라이버 • 디바이스 트리 • 블록 드라이버, 네트워크 드라이버 • 리눅스 멀티미디어				



과정 개설 협의 및 문의



keyseek@naver.com (주)윌텍 류경식 대표



070-8740-2124 (주)윌텍 이정민 실장
010-3342-9323 (주)윌텍 류경식 대표