



Virtual Academy Platform
for Vocational Schools

ECVET

Learning-Unit-Description

Unit	Title of the unit:	Introduction to Programing	
Perquisites	- Basic math		
Work tasks	<ul style="list-style-type: none"> - Learning the basic knowledge in the C++ Programming language (loops, control structures, data types) by practical examples. - Software development Project. 		
Learning outcome	Knowledge:	Skills:	Competence:
	<ul style="list-style-type: none"> - He/she knows how to define the basic processes using UML and state machines. - He/she has a basic understanding of a modern programming language (C/C++) and related concepts. 	<ul style="list-style-type: none"> - He/she is able to analyse the real world process that has to be realised in Software. - He/she is able to realise solutions to basic problems in C++ 	<ul style="list-style-type: none"> - He/she is responsible for process modelling from a real world problem to software solution.
	<ul style="list-style-type: none"> - He/she has a understanding of memory management in C++. 	<ul style="list-style-type: none"> - He/she is able to design save and efficient programs. 	<ul style="list-style-type: none"> - He/she is responsible for the secure implementation of functionality in the C++ programming language.
<ul style="list-style-type: none"> - He she has a understanding of the machine that is running the program and able to recognise if the a program is properly working. 	<ul style="list-style-type: none"> - He/she is able to debug a given software and identify and correct mistakes as well as inefficiencies 	<ul style="list-style-type: none"> - He/she is responsible for bug fixing the software and giving the final release 	

Unit	Title of the unit: Embedded Programming		
Perquisites	<ul style="list-style-type: none"> - Basic math - Basic Programming - Basic Electronics 		
Work tasks	- Practical microcontroller programming skills are taught in project work on a virtualised ATmega32 (VAPVOS)		
Learning outcome	Knowledge:	Skills:	Competence:
	<ul style="list-style-type: none"> - He/she has a detailed understanding of the Microcontroller specific hardware components and the differences between programming on a common computer 	<ul style="list-style-type: none"> - He/she is able to understand the hardware description of the microcontroller and apply this in practical embedded programming - He/she is able to utilise the specific hardware components of the microcontroller 	<ul style="list-style-type: none"> - He/she is responsible for hardware definition and requirement analysis.
	<ul style="list-style-type: none"> - He/she has a understanding of low level programming. - He/she understands how low lever hardware communication works without an operating system involved. 	<ul style="list-style-type: none"> - He/she is able to utilise the full internal functionality of an embedded device. 	<ul style="list-style-type: none"> - He/she is responsible for save and reliable implementation of functionality.
<ul style="list-style-type: none"> - He/she has a understanding of the basic I/O concepts of embedded programming and understands the relations between actions in the program code and electrical actions at the I/O ports of the physical device. 	<ul style="list-style-type: none"> - He/she is able to program analogue and digital Inputs and Outputs and interface external hardware components. 	<ul style="list-style-type: none"> - He/she is responsible for save and reliable interaction between software and external hardware. 	

	<ul style="list-style-type: none"> - He/she has a understanding complex interactions between hard and software recognise if the complete system is properly working. 	<ul style="list-style-type: none"> - He/she is able to debug a given system and identify and correct mistakes in the electrical design, the wiring or the software side interfacing of the electrical components. - He/she can design program code to interface a given electrical circuit. - Can design basic electrical circuits for common tasks. Like pull up, pull down resistors, amplifiers or safety resistors. 	<ul style="list-style-type: none"> - He/she is responsible for bug fixing the complete system design.
	<ul style="list-style-type: none"> - He/she has a understanding of the basic electrical circuits used to interface embedded devices. 	<ul style="list-style-type: none"> - He/she is able to connect switches, signals and basic sensors to the embedded controller, read the signals and properly react on the given input. 	

Unit	Title of the unit: Sensors and Actors		
Perquisites	<ul style="list-style-type: none"> - Basic Math - Basic Electronics - Basic Programming - Basic Physics 		
Work tasks	- Practical experimentation using HomeLab Kit, Lab Work in Virtual and Distance Lab environments		
Learning outcome	Knowledge:	Skills:	Competence:
	<ul style="list-style-type: none"> - He/she knows the basic type of Sensors. - He/she has a basic understanding of the relation between the physical value and the measured electrical signal. 	<ul style="list-style-type: none"> - He/she is able to analyse a process and choose fitting sensors according to the problem on hand. 	<ul style="list-style-type: none"> - He/she is responsible for system design and selection of the right hardware components.
	<ul style="list-style-type: none"> - He/she has an understanding of the physical relations and calculations required to use sensor measurements in C++. 	<ul style="list-style-type: none"> - He/can interface sensors and process real world measurement data in C++. 	

	<ul style="list-style-type: none"> - He/she has an understanding of the physical relations between sensors and measurement software components and able to recognise if the a program is properly working. 	<ul style="list-style-type: none"> - He/she is able to debug a given software and identify and correct mistakes as well as inefficiencies. 	<ul style="list-style-type: none"> - He/she is responsible for bug fixing the software and giving the final release.
	<ul style="list-style-type: none"> - He/she has knowledge of common actors like motors, servos, signals lights, pumps, speakers. - He/she has a understanding of the electrical backgrounds necessary to connect given Actors. 	<ul style="list-style-type: none"> - He/she is able to select physical actors required to solve a certain task and able to conned these to an embedded controller. 	<ul style="list-style-type: none"> - He/she is responsible for process modelling from a real world problem to software solution.

Unit	Title of the unit: Introduction to mobile Robotics		
Perquisites	<ul style="list-style-type: none"> - Basic math - Basic Electronics - Basic Programming - Sensors and Actors 		
Work tasks	<ul style="list-style-type: none"> - Learning the behaviour of complex mechatronic Systems with an practical example of mobile robotics. 		
Learning outcome	Knowledge:	Skills:	Competence:
	<ul style="list-style-type: none"> - He/she knows best practice methods for robot navigation using sensor data. - He she has a basic understanding of sensor data based safety features. - He she has a understanding of actors and hardware components used in robotic applications (motors, servos, motor drivers). 	<ul style="list-style-type: none"> - He/she is able to design a robotic system able to navigate in its environment. - He/she is can identify hardware components required for robot interaction. 	<ul style="list-style-type: none"> - He/she is responsible for process modelling from a real world problem to software solution.
	<ul style="list-style-type: none"> - He/she has a understanding of software components and strategy's solving common robotic problem sets in C++. 	<ul style="list-style-type: none"> - He/she is able to design save and efficient programs for autonomous robotics. 	<ul style="list-style-type: none"> - He/she is responsible for the secure implementation of functionality in the C++ programming language.

	<ul style="list-style-type: none">- He she has a understanding of the robotic Hardware and Software interaction and able to recognise if the a program is properly working.	<ul style="list-style-type: none">- He/she is able to debug a given software and identify and correct mistakes as well as inefficiencies	<ul style="list-style-type: none">- He/she is responsible for bug fixing the software and giving the final release
--	---	--	--