

# Programming For Engineers

## Course Guidelines

by

Wan Azhar Wan Yusoff<sup>1</sup>, Ahmad Fakhri Ab. Nasir<sup>2</sup>  
Faculty of Manufacturing Engineering  
wazhar@ump.edu.my<sup>1</sup>, afakhri@ump.edu.my<sup>2</sup>



# Course Synopsis

This course introduces the parallel/serial interfacing techniques between PC and external circuit built with the components such as LEDs, motors (DC/stepper), thermometer etc. using C/C++ programming language. In addition, the intermediate level of programming techniques such as pointers, dynamic memory allocation, data structures, and graphical user interface (WIN32 GUI) are also introduced to fit the purpose. By the end of semester, the students apply the interfacing techniques in a mechatronics-based project.



# Course Synopsis

By the end of the semester, students should be able to:

CO 01: Apply concepts of pointers, data structures and logical bitwise.

CO 02: Develop graphical user interface (WIN32 GUI).

CO 03: Construct an integration software with electrical devices/components and mechanical system.

CO 04: Orally present and collaborate effectively in a group on a mechatronics-based project.



# Assessment Summary

Methods	%	CO 01	CO 02	CO 03	CO 04	Assessed Competency	Program Outcome
WIN32 GUI Programming	20%					Competency of GUI	<b>PO 03:</b> Design and develop solution
Project Report & Presentation	20%					Delivery Q & A and group project report	<b>PO 09:</b> Communication
Project System	15%					Group project system functioning	<b>PO 05:</b> Modern tools usage
Test I	15%					Pointer and memory	<b>PO 01:</b> Knowledge
Test II	15%					Data structure	<b>PO 01:</b> Knowledge
Test III	15%					Logical bitwise	<b>PO 01:</b> Knowledge
<b>Total</b>	<b>100%</b>						



# Assessment Summary

This course will measure (and develop) your abilities not only in cognitive domain but also the psycho motor and affective domains.

## Course Project

In a group of several students, a mechatronic system is designed and developed. Each student will play a role in the development process. Examples of mechatronic system are four-bar linkage (dynamic mechanism) or a dynamic (Newton's Law) experiments. We expect a prototype but the emphasis is more on programming and electronics rather than mechanical.

## Written Test

Written tests, as usual are open-book, open-note sessions. We expect knowledge and understanding of intermediate C/C++ programming elements such as pointer, user-defined data type (struct) etc. Other than understanding, we expect a small program design to achieve specific purpose. we use tests to differentiate individual competency.



# Course Syllabus

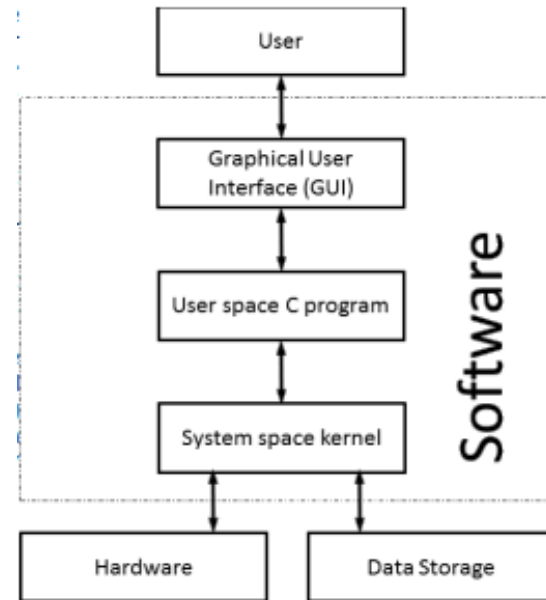
Week	Topic	Outcomes
01 – 02	Beyond basic data type I: Using pointer and memory	Simple C program.
03 – 04	Beyond basic data type I: Using struct	Simple C program.
05	Logical bitwise (bit field) programming	Simple flags programming.
06	Organizing program files, compiling, linking and library.	Simple C program.
07 – 08	User interfacing	Windows GUI programming
09	USB Interfacing I	Arduino LED ON/OFF, ADC and DAC. Temperature measurement, and potentiometer.
10	USB Interfacing II	Arduino frequency READ/WRITE to stepper motor and DC motor with simple encoder.
11 – 14	Mechatronic Project	Report and Presentation





# Course Syllabus

We will use the three-tier software model: GUI, engine and database. We will use WIN32 GUI, C/C++ program engine and text file. We will apply USB interface programming – an interface with microcontroller (Arduino UNO) to perform DAC – Digital to Analog Converter (executing a DC motor), ADC – Analog to Digital Converter (reading temperature sensor) and pulse reading (encoder) and writing (stepper motor drive). Before that, we will understand how data are organized – mainly the struct. Because of that, we need to understand pointer – an efficient way to handle data.



# Course Syllabus

The above showed diagram is a typical mechatronic system. GUI is the part of software that allows users to interact with program. We can develop GUI from many technology such as Qt, WIN32, Gtk etc. User space is our C/C++ program. System space is the operating system such as Windows, UNIX, Linux, etc. Compiler is the software that convert text program into object modules. Linker combines object modules to produce executable. Library is a module. It can be static and dynamically linked. Application is executable program. We interface with hardware through I/O port. We can use USB, parallel port, PCI port etc. We interface with data storage using operating system call. We will write and read our data to hard drive either in text or binary format. We will program in C/C++ language but mainly C language. We will compile using gcc compiler.





# Learning References

1. B. A. Forouzan and R. F. Gilberg (2007). Computer Science: A Structured Programming Approach Using C. Third Edition. Cengage Learning.
2. C. S. Horstmann and T. A. Bud (2017). Big C++. Third Edition. John Wiley & Sons.
3. J. Katupitiya and K. Bente (2006). Interfacing with C++: Programming Real World Applications. Berlin Heidelberg.

