BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
K. K. BIRLA GOA CAMPUS and HYDERABAD CAMPUS
II SEMESTER 2019-2020
COURSE HANDOUT (PART II)

Date: 3/1/2020

In addition to Part I (General Handout for all courses appended to the timetable) this portion gives further specific details regarding the course.

Course No. **:** CS G523

Course Title **:** Software for Embedded Systems

Instructors : NEENA GOVEAS, S. Ershad Ahmed

1.1 Objective:
Introduce the students to the issues and challenges in developing software for embedded systems. Educate them in formal modeling, design and development methodologies. Expose them to tools and techniques used for developing such software.

1.2 Scope:
The course will review basics of Embedded and Real-time systems and generic software processes. The primary focus will be on identifying specific issues related to embedded software development within the usual phases of software development cycle – requirements, design, implementation and testing. A number of illustrative examples will be used as ad-hoc case studies for different phases to bring out the issues and challenges in the process. Specific technologies, operating environments and development tools will be also used as part of the practical aspects of the course.

1.3 Course Description:
Embedded Systems; Software issues in Embedded Systems; Software Development Process; Requirements Analysis – Model driven systems design - UML – Architectural Design, Design Patterns, Detailed Design. Networking Protocols; Implementation, Run-time environments and Operating systems for embedded software. Testing – Methodologies, Test Cases. Application development in Embedded Systems- Using Robot Operating System.

2.1 Text Book:

T1 Bruce Powel Douglass, "Real – Time UML: Advances in the UML for Real – Time Systems", 3rd Edition, 2004.
T2 Dive into Python, https://www.python.org/
T3 Programming Robots with ROS: A Practical Introduction to the Robot Operating System by Brian Gerkey, Morgan L. Quigley, and William D. Smart. https://www.ros.org/

2.2 Reference books:

R1 Frank Vahid and Toby Givargis, "Embedded System design", Wiley.

R2 James Rumbaugh, Ivar Jacobson and Grady Booch, "The Unified Modeling Language Reference Manual", 2nd edition, Pearson Education, 2005.

R3 Luciano Lavagno, Grant Martin and Bran V Selic, "UML for Real: Design of Embedded Real-Time systems", Kluwer Academic Publishers, 2003.

R4 http://www.temida.si/~bojan/IPIT_2014/literatura/UML_Reference_Manual.pdf

R5 http://wiki.ros.org/Books

R6 https://processing.org/

3.1 Lecture Modules & Learning Objectives

Module	Title	Learning Objectives
1	Introduction to Embedded Systems	Review of basics and understanding issues in Embedded Systems
2	Embedded System Modeling	Techniques for Modeling Embedded Systems
3	Software Design of a Product	Software Development Methodologies and phases
4	Software Design for Real – Time / Embedded Systems	Understanding the Software design process by identifying and analyzing requirements for embedded systems software
5	Design, implementation and testing	Understanding high level architectural framework(s) and design patterns applicable for embedded software. Understanding detailed design issues, implementation issues, Testing methodology for embedded system.
6	Operating system and run time environments for Robots	Robot Operating System and its use for developing applications

3.2 Lecture Schedule

Mod #	Lect #	Topics	References
1	**1 – 2**	**Introduction to embedded system**	
	1	Introduction to embedded and real – time system	R1
	2	Characteristics of software for embedded and real – time systems	T1and R1
2	**3 – 7**	**Object Oriented modeling**	
	3 – 6	Object oriented programming using Python	T2 and Class notes
	7	Embedded systems and Object oriented approach-	Class notes
3	**8 – 16**	**Software design of a product**	
	8 – 10	Object oriented modeling and design with UML	T1, R2 and R3

	11 – 13	UML, UML diagrams, use cases and scenarios – Identification, Details and Diagrams	T1, R2 and R3
	14 – 16	Analysis – Object Discovery & Identification, Objects Associations & Attributes, Classes and Relationships, State and Behavior, State charts and Scenarios	T1, R2 and R3
4	**17 – 23**	**Software development for Real – time / Embedded/ networked systems**	
	17-21	Processing programming environment	R6
	22 – 23	Processing and Arduino communication	R6
5	**24 – 26**	**Operating systems and Run-time Environments in Embedded system**	
	24 – 26	Robot Operating System: Introduction	T3, R5
6	**27 – 35**	**Embedded systems- Interactions with environment using sensors**	
	27	Publisher-Subscriber using Python	R5
	28 – 29	Navigation using ROS- Use of sensors for distance	
	30 – 35	Path planning using ROS- Issues	
7	**36 - 37**	Modeling Embedded Devices	
	38	UML Mapping - Revisit of software engineering to make it suit for Real – time / Embedded system	R8
	39 – 40	Models for Embedded Software design	R9

4.1 Evaluation Scheme:

Components	Weightage	Date & Time	Remarks
Test I	30%	To be announced later	Closed Book
Assignment/lab project	25%	Using System-on-Chip (SOC) platform (Embedded device based. Confirm the suitability of device you plan to use.)	Open Book
Presentation	5%	Defending the project design	Open Book
Comprehensive Examination	40%		Closed Book

4.2 Projects:

- Project will include the following components:
  (a) Problem Identification, Specification, and Use Cases.
  (b) Requirements Analysis, State-charting.
  (c) Modular Design/Detailed Design
  (d) Prototype Implementation as part of the Labs and classes

- Students should present their work as presentation seminars and should attend others presentations.
- Projects will be evaluated individually through presentation / demonstration.
- Projects are to be completed in time with no postponements.

4.3 Malpractice Regulations:

1. Any student or team of students found involved in mal practices in working out projects will be awarded negative marks equal to the weightage of that project and will be blacklisted.
2. Any student or team of students found repeatedly – more than once across all courses – involved in mal-practices will be reported to the Disciplinary Committee for further action. This will be in addition to the sanction mentioned above.
3. A mal-practice - in this context - will include but not be limited to:
   - Submitting some other student's / team's solution(s) as one's own;
   - Copying some other student's / team's data or code or other forms of a solution;
   - Seeing some other student's / team's data or code or other forms of a solution;
   - Permitting some other student / team to see or to copy or to submit one's own solution;
   - OR other equivalent forms of plagiarism wherein the student or team does not work out the solution and/or uses some other solution or part thereof (such as downloading it from the web).
4. The degree of mal-practice (the size of the solution involved or the number of students involved) will not be considered as mitigating evidence. Failure on the part of instructor(s) to detect mal-practice at or before the time of evaluation may not prevent sanctions later on.

5. **Chamber Consultation Hour**: To be announced in the class.

6. **Notice**: Notice concerning this course will be displayed on LMS

7. **Makeup Policy**:

- Permission of the Instructor is required to take a make-up
- Make-up applications must be given to the Instructor personally.
- A make-up test shall be granted only in genuine cases wherein the Instructor's judgment - the student would be physically unable to appear for the test.
- In case of an unanticipated illness preventing a student from appearing for a test, the student must present a Medical Certificate from BITS medical centre.
- Requests for make-up for the comprehensive examination – under any circumstances – can only be made to Instruction Division.

Instructor-in-Charge
CS G523