SYLLABUS

1. **Course name:** Embedded Systems Lab
2. **Course code:** ESPR427064
3. **Credits:** 2 credits (0/2/4) (0 theoretical credits, 2 practical credit).
   Duration: 15 weeks (6 main periods and 12 self-study periods)/week).
4. **Instructors:**
   a. Primary instructor: Le Minh Thanh, M.Eng.
   b. Secondary instructors:
   - Truong Quang Phuc, M.Eng.
   - Truong Ngoc Ha, M.Eng.
5. **Course conditions:**
   a. Prerequisites: N/A.
   b. Corequisites: Embedded Systems.
6. **Course Description:**

   In this class, the fundamentals of embedded system hardware and firmware design will be explored. Issues such as embedded processor selection, hardware/firmware partitioning, glue logic, circuit design, circuit layout, circuit debugging, development tools, firmware architecture, firmware design, and firmware debugging will be discussed. The STM32F4, a very popular microcontroller family, will be studied. The architecture and instruction set of the microcontroller will be discussed, and a wirewrapped microcontroller board will be built and debugged by each student. The course will culminate with a significant final project which will extend the concepts covered earlier in the course. Learning may be supplemented with periodic guest lectures by embedded systems engineers from industry. Depending on the interests of the students, other topics may be covered.
7. **Course Goals**

Goals	Goal description *This course provides students:*	ELOs
G1	Surveying embedded development tools (IDEs), Win CE, Python, Tkinter, Raspberry Pi	01(M), 02(M)
G2	Analysis of hardware design on embedded systems.	03 (H)
G3	Application programming on embedded systems using Keil C, CubeMX programming language.	08 (M), 09(L)
G4	Ability to communicate in technical environment, teamwork and read communication documents.	06 (H)
G5	Ability to perform steps to design a embedded system.	07 (M)
G6	Ability to measure, test, analyze, calculate, design and assembly embedded systems.	10 (H), 11 (M)

***** *Note: H: **H**igh; M: **M**edium; L: **L**ow*

8. **Course Learning Outcomes - CLOs:**

CLOs	Description *After completing this course, students can have:*	ELOs

G1	G1.1	Know how to use the STM32F4 kits, Ras Pi kits.	01
	G1.2	Ability to present block diagrams of STM32F4 kits,. Ras Pi kits	01
G2	G2.1	Ability to analyze and fix errors in a STM32F4 kits, Ras Pi kits .	02
	G2.2	Ability to analyze and fix errors when connecting blocks in a system.	02
G3	G3.1	Ability to use Keil, CubeMX softwares to programming STM32F4 kits, Ras Pi kits	03
G4	G4.1	Ability to perform operation of the experimental kits.	08, 09
	G4.2	Ability to solve the system' problem.	06, 06
	G4.3	Reading digital datasheet.	07
G5	G5.1	Ability to perform steps to design a communication system.	07, 11
G6	G6.1	Ability to design and assembly UART communication, SPI communication, TIM, interup …	10,11

9. Study materials:

a. Textbooks:

[1]. STM32F4 Labs.

b. References:

[2]. Datasheets stm32f4 discovery

10. Student Assessments:

a. Grading points: 10

b. Planning for students assessment is followed:

Type	Contents	Line time	Assessment techniques	CLOs	Rates (%)
Report					**50**
P	Embedded Operating System.	Week 6	Report by individual / group	G1, G4, G5	10
P	Embedded software Programming in Python	Week 10	Report by individual / group	G1, G4, G5	20
P	Ras. Pi based-Embedded system prototyping	Week 14	Report by individual / group	G1, G4, G5	20
Test 1					**50**
M	Paper: Win CE, Gui Python, Raspberry PI.	Week 15	Individual students.	G2, G6	20
Test 2					**30**
M	Lab: Win CE, Gui Python, Raspberry PI.	Week 15	Assembling, measuring.	G2, G6	30

** Note: Q: **Q**uiz; H: **H**omework; P: **P**roject; M: **M**idterm Exam; F: **F**inal Exam;*

11. Course details:

Week	Contents	CLOs

1	***Lab 1.* Introduction** *(0/6/12)*	
	Teaching contents: *(6)* 1.1. Course overview, expectations, logistics, processes, syllabus, FAQ, and prerequisite material. 1.2. Embedded systems descriptions, definitions, and vocabulary. Design Engineer's notebook. 1.3. Design considerations and requirements, processor selection and tradeoffs 1.4. Overview of board development process, wire wrapping vs. soldering. **Teaching methods:** + Theoretical lectures. + Previous operation.	G1.1 G1.2
	Self-study contents: *(12)* + Review relevant knowledge.	
2	***Lab 2.* Configure GPIO for LED toggling** *(0/6/12)*	
	Teaching contents: *(6)* 2.1 Learn how to setup pin and GPIO port in CubeMX 2.2 Configure GPIO pin in CubeMX and Generate Code 2.3 Add in to project HAL_Delay function and HAL_GPIO_Toggle function. **Teaching methods**: + Theoretical lectures. + Previous operation.	G1.2 G3.1 G3.2 G5.1 G6.1
	Self-study contents: *(12)* 2.4 Verify the correct functionality on toggling LED.	
3	***Lab* 3 Configure EXTI which turns on LED.** *(0/6/12)*	
	Teaching contents: *(6)* 3.1.Configure GPIO and EXTI pin in CubeMX and Generate Code 3.2.Add into project Callback function and function which turn on led. **Teaching methods**: + Theoretical lectures. + Previous operation.	G1.2 G3.1 G3.2 G5.1 G6.1
	Self-study contents: *(12)* Verify the correct functionality by pressing button which turns on LED	
4	***Lab 4.* Use SLEEP mode with EXTI** *(0/6/12)*	
	Teaching contents: *(6)* 4.1. Use project from EXTI lab 4.2. Learn how to setup the SLEEP in HAL, and which events can wake up MCU. **Teaching methods**: + Theoretical lectures. + Previous operation.	G1.2 G2.1 G4.1, G4.2 G5.1 G6.1
	Self-study contents: *(12)* + Verify the correct functionality by measuring consumption .	G3.1 G5.1 G6.1

5	***Lab 5.*** **Data transfer over DMA** *(0/6/12)*	
	Teaching contents: *(6)* 5.1 Use CubeMX and Generate Code with DMA 5.2 Learn how to setup the DMA in HAL 5.3 Verify the correct functionality by comparing transferred buffers. **Teaching methods**: + Theoretical lectures. + Previous operation.	G2.2 G4.1 G6.2
	Self-study contents: *(12)* + Data transfer over DMA with interrupt.	G4.3 G5.1 G6.2
6	***Lab 6.*** **Creating a Project with Windows CE** *(0/6/12)*	
	Teaching contents: *(6)* 6.1. Applications for Microsoft Foundation Classes (MFC) for Windows CE 6.2. Objects for the MFC Component Object Model (COM) **Teaching methods**: + Theoretical lectures. + Previous operation.	G1,G4, G5
	Self-study contents: *(12)* COM projects for the Active Template Library (ATL) for Windows CE.	
7	**REPORT 1**	G2, G6
8	***Lab 7.*** **Simple UART communication** *(0/6/12)*	
	Teaching contents: *(6)* 7.1 Configure UART in CubeMX and Generate Code 7.2 Learn how to send and receive data over UART without interrupts 7.3 Verify the correct functionality. **Teaching methods**: + Theoretical lectures. + Previous operation.	G2.2 G3.1 G4.1 G6.2
	Self-study contents: *(12)* + Use UART with interrupt	G3.1 G4.3 G6.2
9	***Lab 8.*** **GUI IN PYTHON** *(0/6/12)*	
	Teaching contents: *(6)* 8.1. Import the *Tkinter* module. 8.2. Create the GUI application main window. 8.3. Add one or more of the above-mentioned widgets to the GUI application. **Teaching methods**: + Theoretical lectures. + Previous operation.	G2.2 G3.1 G4.1 G6.2

	Self-study contents: *(12)* + Enter the main event loop to take action against each event triggered by the user	G3.1 G4.3 G6.2
10	**REPORT 2**	G2, G6
11	***Lab 9.*** **Simple SPI communication** *(0/6/12)*	
	Teaching contents: *(6)* 9.1. Configure SPI in CubeMX and Generate Code 9.2. Learn how to send and receive data over SPI without interrupts Verify the correct functionality. **Teaching methods**: + Theoretical lectures. + Previous operation.	G2.2 G4.1 G4.2 G6.2
	Self-study contents: *(12)* + Use SPI with interrupt.	G5.1 G6.2
12	***Lab 10.*** **TIM with interrupt** *(0/6/12)*	
	Teaching contents: *(6)* 10.1. Configure TIM in CubeMX and Generate Code 10.2. Learn how start timer and handle interrupt 10.3. Verify the correct functionality. **Teaching methods**: + Theoretical lectures. + Previous operation.	G2.2 G4.1 G4.2 G6.2
	Self-study contents: *(12)* + Design 16-input adder.	G5.1 G6.2
13	***Lab 11.*** **RASPBERRY PI 3** *(0/6/12)*	
	Teaching contents: *(6)* 11.1. Introduction to the Raspberry Pi and the gear you will need to get started 11.2. How the GPIO pins are configured on the Raspberry Pi 3. **Teaching methods**: + Theoretical lectures. + Previous operation.	G1, G4, G5
	Self-study contents: *(12)* + How to control two LEDs from two push buttons using Python. This requires us to use GPIO pins as inputs and outputs.	
14	**REPORT 3**	G2, G6
15	**TEST**	G2, G6

12. Learning ethics:

- Home assignments and projects must be done by the students themselves. Plagiarism found in the assessments will get zero point.
- Students who attend less than 80% or do not complete 80% of homework will be banned.

13. First approved date: 01/01/2012

14. Approval level

Dean	**Department**	**Instructor**
Nguyen Minh Tam, Ph.D	Nguyen Ngo Lam, M.Eng	

15. Syllabus updated process:

1st time: Updated content dated: 15/01/2014 *Contents:*	**Instructor:** **Head of department:** Vo Minh Huan, Ph.D
2nd time: Updated content dated*:* 15/01/2016 *Contents:*	**Instructor:** **Head of department:** Phan Van Ca, Ph.D
3rd time: Updated content dated*:* 06/05/2017 *Contents:*	**Instructor:** Le Minh Thanh, M.Eng **Head of department:** Phan Van Ca, Ph.D