CO326 Project:

Please note that every year a new project is given for CO326 and it is an open-ended project and does not have a readymade solution. It allows us to learn state-of-the-art and also think out of the box. It is a learning exercise for the entire team. Therefore please feel free to discuss among yourself as well as with lecturers and instructors to propose new products and solutions.

2023 Project Description: Development of IoT modules using ESP

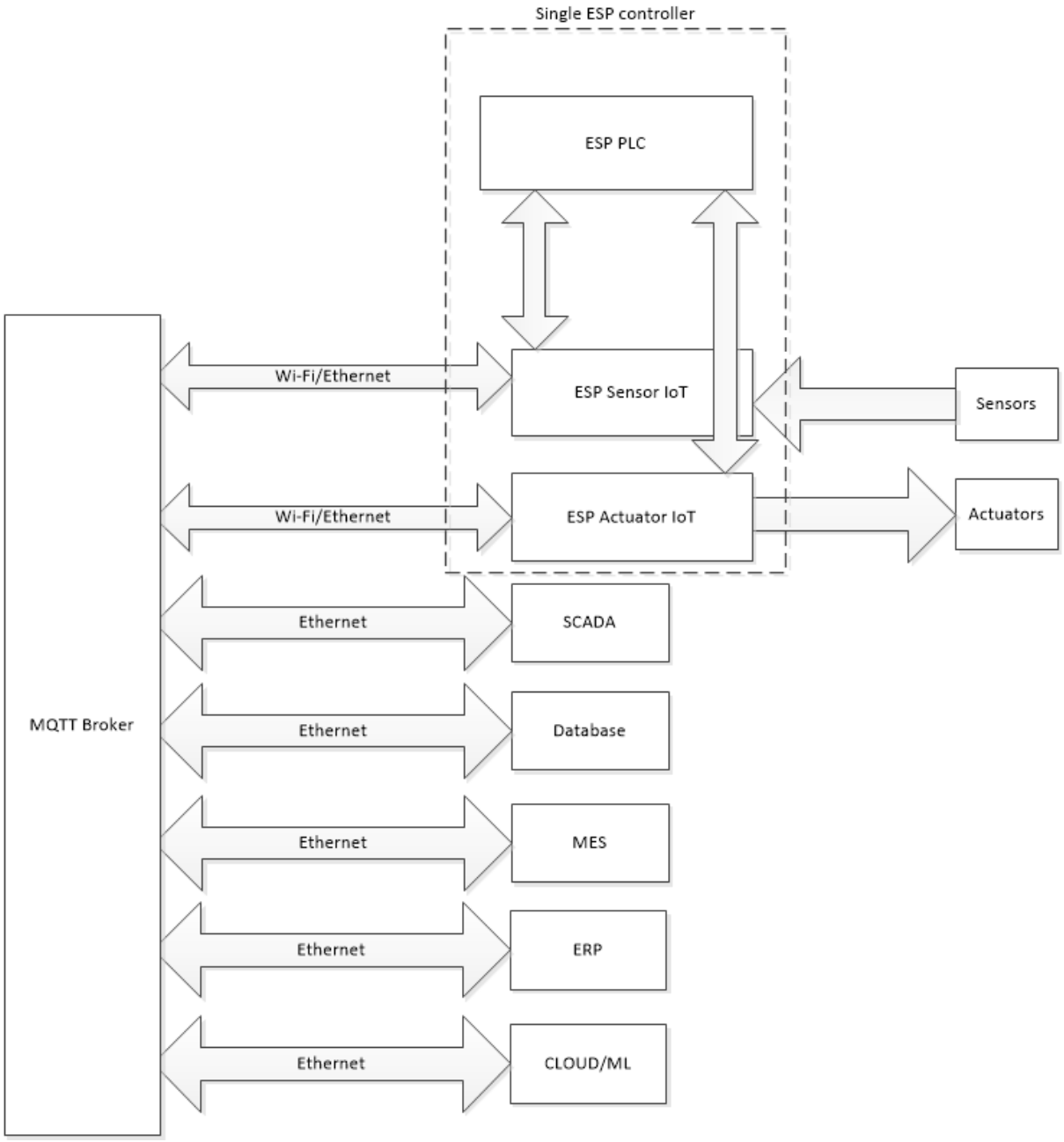
Version 3: Revised on 06-04-2023

Learning Objectives

1. Learn Industry 4.0 concepts
2. Develop an IIoT smart sensor and smart actuator
3. Implement a broker that connects IIoT device
4. Develop a simple automation project with SCADA and History Database

Project Description

ESP8266/ESP32 has a large number of I/O capabilities. The CO326 project aims to develop IoT modules using ESP8266/ESP32. The basic arrangement is shown in Fig.1. In this design ESP8266/ESP32 communicates with sensors or actuators. The IIoT sends collected data from the sensor to a central MQTT Broker. The actuator consumes data in the MQTT Broker. A computer connected to the MQTT Broker presents the output on a SCADA. The process control functions are implemented on SCADA. Also, the historical data is stored in a database.



Note: Industry 4.0 applications: MES=Manufacturing Execution System, ERP=Enterprise Resource Planning, CLOUD/Machine Learning Applications

Fig 1. The basic arrangement of control and communication system

The standard sensors and actuators available in the market are to be converted to smart sensors and actuators with the help of ESP controllers. Each smart sensor or actuator should have a controller. The control function is implemented on a PLC. However as we do not have a large number of ESP controllers, the codes of sensors, actuators and PLC are implemented on the single ESP controller as shown in Fig1.

The codes for sensors, actuators, control sequence and communication should be written as separate functions on the same ESP controller. The data transfer between each function happens through variables passed between functions.

The variables outside the functions are defined as variables in Unified Name Space. The unified namespace identifies the data and the parameters in the form of Enterprise/Site/AREA/Line. Following the same format, the names used in the project should be UoP_CO_326_E18_GrNo_Sensor or Actuator Name_Data or Parameter Name (Slashes replaced by underscore).

The ESP reads the sensor data and sends it to the MQTT Broker. The data consumed by the broker controls the actuator. Also, all parameters of the sensor and actuator can be set by the values written to the MQTT broker. The sensors and actuators should be configured only by changing the parameters using the broker and without changing the programme.

Each group should use a minimum of one sensor and one actuator. Preferably, a type of sensor and actuator may be used by only one group. The basic sensors and actuators should be improved with additional parameters required for Industry 4.0. For example, a simple temperature sensor could have functions to give the temperature in Centigrade or Fahrenheit. Additional parameters such as the date of installation, the number of readings obtained, and the sensor parameter such as the gain, offset etc. may be included. A motor could have the number of running hours, start stops etc which needs for taking ERP-level decisions. The sensor data of different groups may be shared via the broker.

Wi-Fi will be used to connect to the Internet for long-distance communication. It is recommended to use EMQX or Mosquitto as MQTT Broker, MYSQL as database and NodeRed or ScadaBR as SCADA. The possibility of providing a central broker and a database is being considered. Therefore, it is recommended to start developing functions for ESP smart devices first.

The controller time stamps the collected data accurately to 1 ms before sending it to the MQTT broker. The MQTT Broker time stamps the data at the time of receiving. The SCADA and Database store the data with the time it receives the data. SCADA will display the data in graphical format with the time taken to transmit data from the controller to the SCADA. The time-stamped at the controller, Broker and the SCADA/Database gives the ability to analyse the round trip communication time. All clocks are synchronised to an international time server.

The project will be evaluated based on the hardware and software implementation, project report and final presentation. The project report should provide sufficient implementation details including the hardware, software, protocols, designs and references. A GitHub account should be maintained and include all the above submission requirements including the references.

Each project group will comprise a maximum of three members. There will be regular meetings to discuss the progress of the project. In addition, the groups are free to meet the lecturers whenever necessary.

Marking Scheme

Description		Marks
Separation of functions and use of variables to pass data Max 5 per function based on the complexity of the function	Sensors (Digital/Analogue)	20
	Actuators (digital/analogue)	20
	Control logic	10
Sending data to MQTT broker with Unified Naming		15
Storing data in a database		10
Implementation of a SCADA	Display Digital	4
	Display Analogue	4
	Control Digital	4
	Control Analogue	4
	Present Event List	4
	History Event List from the database	5
Additional developments	Bonus marks Max	10

Required Submissions

1. Project presentation with circuits and GitHub link to source code
2. Photos of hardware
3. Video of demonstration