MIT | Academy of Engineering

Lab Manual

Course Name - Engineering Informatics

Course Code – 2207232
Course Credits – 04
Course Type - Program Core Course

Class & Branch – SY BTECH [All Branches]

MIT Academy of Engineering Alandi Pune

Revision 2022

MIT Academy of Engineering AN AUTONOMOUS INSTITUTE	**INDEX & CERTIFICATE**	
	ACADEMIC YEAR	
Alandi (D), Pune – 412105	**SEM/TRI**	
DEPARTMENT OF ……………………ENGG.	**CLASS & BLOCK**	

This is to certify that Mr/Ms ---Roll No ---------------------has successfully completed the experiments for the course **Engineering Informatics** for academic year 20_________.

Sign
Student

Sign
Course instructor

Module No. 2	**Title:** Data Acquisition , Storage and Retrieval Systems using Arduino / Raspberry Pi	14 Hours

Deploy an IoT based automation system in the sector of Smart Agriculture, Smart Home Automation, Smart City, Disaster management, Military applications, Industrial automation etc and develop a IoT based system which will include following,

a. Identification of an interdependent elementary data items which have facts and figures
b. Data collection through sensors
c. Processing using Arduino / Node MCU
d. Data Storage on cloud through Thinkspeak.
e. Device control using mobile Apps or through Web pages.
f. Data storage in file format.

Problem Statement –

Deploy an IoT based automation system in the sector of Smart Agriculture, Smart Home Automation, Smart City, Disaster management, Military applications, Industrial automation etc and develop a IoT based system

It is expected that students will implement any one system from the given examples or whatever real life application possible.

Objectives –

- Identification of components
- Circuit design & development using PCB
- Connections of various components
- Circuit Analysis & Testing

Outcome –

After the successful completion of experiment students will be able to,

- Identify the components required to implement the automation system [Understand]
- Design the circuit using PCB [Create]
- Analyze the circuit for testing purpose [Analyze]

Theory –

Deploying IoT (Internet of Things) applications involves selecting suitable processors to run the application code and handle communication with connected devices and sensors. The choice of processor depends on the specific requirements of the IoT application, such as power consumption, processing capabilities, memory, and connectivity options. Here's a general overview of the steps involved in deploying an IoT application using a processor:

1. Define Application Requirements: Start by clearly defining the requirements of your IoT application. Consider factors such as the number of connected devices, data processing needs, communication protocols (e.g., Wi-Fi, Bluetooth, Zigbee, cellular), and power constraints.

2. Select the Right Processor: Based on your application requirements, choose a suitable processor. There are various types of processors available, including microcontrollers (MCUs), microprocessors (MPUs), and system-on-chip (SoC) solutions. Microcontrollers are typically used for low-power and resource-constrained applications, while microprocessors and SoCs offer more processing power and capabilities.
3. Develop the Application Code: Write the IoT application code using the programming language and development tools that are compatible with the selected processor. Common programming languages for IoT applications include C, C++, Python, Java, and JavaScript.
4. Connect Devices and Sensors: Integrate the IoT devices and sensors with the selected processor. This may involve setting up communication protocols and drivers to establish connectivity and data exchange between the processor and the devices.
5. Set Up Communication: Implement the required communication protocols to enable data exchange between the IoT devices, sensors, and the central processing unit. This might involve configuring Wi-Fi, Bluetooth, or other communication modules on the processor.
6. Security Implementation: Implement security measures to protect data and ensure secure communication between the devices and the processor. This may include encryption, authentication, and secure boot mechanisms.
7. Test and Optimize: Thoroughly test the IoT application on the selected processor to ensure its reliability, performance, and functionality. Optimize the code and the communication to achieve the desired level of efficiency and responsiveness.
8. Deploy the IoT Application: Once the IoT application is fully tested and optimized, deploy it on the selected processor and connect it to the IoT devices and sensors in the target environment.
9. Monitor and Maintain: Continuously monitor the IoT application's performance, data flow, and security. Regular maintenance and updates may be necessary to address any issues or add new features.
10. Scale and Expand: As the IoT application grows and new requirements arise, consider scalability and expansion. Depending on the processor's capabilities, you might need to upgrade the hardware or explore cloud-based solutions for more extensive IoT deployments.

It's important to note that IoT application deployment can vary significantly based on the complexity of the application, the chosen processor, and the specific use case. Proper planning, testing, and optimization are essential for successful IoT application deployment using a processor.

Procedure –

After the selection of IoT based Automation system, follow the steps given below,

a. Identification of an interdependent elementary data items which have facts and figures – Enlist the various parameters which is required to be measure with the help of system implemented.

    1. Distance:

    Required for accurate scanning to measure the distance between the scanner and the object.

    2. Angle/Position:

    To determine the position of the scanner relative to the object being scanned.

3. Light Intensity:

Necessary for controlling the intensity of light sources used for scanning and ensuring optimal scanning conditions.

4. Image Data:

The raw image data captured by the scanner's cameras for processing into a 3D model.

5. Color Data:

If the scanner captures color information, it's important for accurate texture mapping in the resulting 3D model.

6. Temperature:

Monitoring temperature can be important to ensure the components of the scanner are operating within acceptable ranges and to account for any temperature-related effects on measurements.

7. Velocity/Speed:

Useful for assessing the movement of the scanner during the scanning process.

8. Power Consumption:

Monitoring power consumption can help in optimizing the scanner's energy usage and ensuring it operates within specified limits.

9. Calibration Data:

Information related to the calibration of the scanner, such as calibration parameters and coefficients.

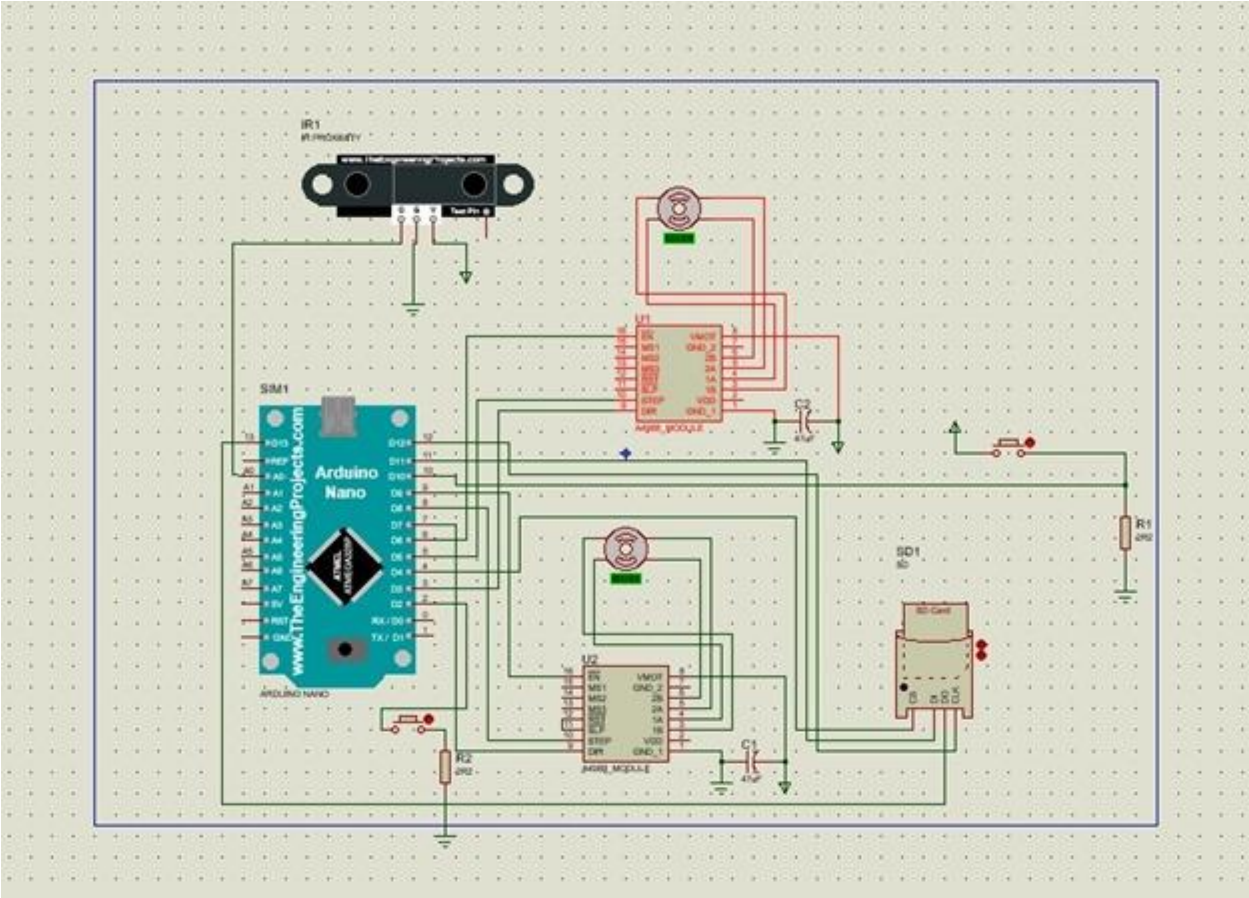
10. Error/Quality Metrics:

Metrics to assess the quality and accuracy of the scanned data, such as point cloud density, resolution, and error rates.

11. Connectivity Status:

Information on the status of connectivity between the scanner and the controlling device (e.g., Wi-Fi, Bluetooth).

b. Design Circuit diagram –



c. Data collection through sensors - Prepare a table of observation and record the parameters and values observed with the help of sensors.

Sr. no	Name of Sensor	Parameter sensed	Reading 1	Reading 2	Reading 3	Reading 4	Average Value
1	Depth	Height of Object	120 cm	118 cm	121 cm	119 cm	119.5 cm
2	Distance	Distance	80 cm	78 cm	81 cm	79 cm	79.5 cm
3	Position	Angular Position	45 degrees	44 degrees	46 degrees	45 degrees	45 degrees
4	Imaging	RGB Color	Blue	Red	Green	Yellow	N/A
5	Proximity	Proximity	Detected	Detected	Detected	Detected	Detected
6	Force/Torque	Force	10 N	9.8 N	10.2 N	10 N	9.5 N

7	Rotation	Angular Position	90 degrees	91 degrees	89 degrees	90 degrees	90 degrees
8	Accelerometer	Acceleration	2 m/s^2	1.8 m/s^2	2.2 m/s^2	2 m/s^2	2 m/s^2
9	Inertial Measurement Unit	Angular Velocity	100 deg/s	98 deg/s	102 deg/s	100 deg/s	100 deg/s
10	Flex	Flexion	30 degrees	29 degrees	31 degrees	30 degrees	30 degrees

d. Processing using Arduino / Node MCU – Write your observation for data Processing by the Microprocessor or Microcontroller used in your system.

Processing data using Arduino or NodeMCU presents certain limitations due to their relatively low processing power and memory constraints. These microcontrollers are best suited for simpler tasks and real-time operations, with Arduino relying on a simplified C/C++ programming language and NodeMCU supporting Lua scripting. Despite these limitations, both platforms offer extensive peripheral support and benefit from large communities, providing access to libraries and forums for development assistance. NodeMCU, with its integrated Wi-Fi capabilities, is particularly well-suited for IoT applications, facilitating data processing and communication with remote servers or devices. However, developers must carefully optimize their code to manage resource limitations and ensure efficient data processing.

e. Data Storage on cloud through Thinkspeak – Describe the process of data storage on Cloud through Thinkspeak or any other software used in your project. Also explain in detail the procedure of cloud interfacing.

For the 3D scanner project, data from sensors capturing object dimensions is formatted by the microcontroller (like Arduino or NodeMCU) and sent to ThingSpeak's cloud servers via an internet connection. ThingSpeak's APIs facilitate seamless data transmission and storage. This stored data can then be remotely accessed for analysis and visualization, enhancing the efficiency of 3D scanning and data management.

f. Device control using mobile Apps or through Web pages. – Give the complete details related to Mobile Apps development or through Web pages.

For the 3D scanner project, mobile app development or web page integration allows users to remotely control and monitor the scanning process. The mobile app/web page would include features such as initiating scans, adjusting scan parameters (e.g., scan amount, step delay), and viewing real-time scan progress. Development involves creating a user-friendly interface using tools like Android Studio for mobile apps or HTML/CSS/JavaScript for web pages.

Communication between the app/web page and the scanner system is typically achieved via Wi-Fi or Bluetooth, with protocols like MQTT or HTTP used for data exchange. This integration enhances user convenience and accessibility, enabling seamless control and management of the 3D scanning process from anywhere with internet connectivity.

g. Data storage in file format - Give the procedure of data storage on cloud, description of software used, connection establishment steps. Write the sample values in given observation table.

For the 3D scanner project, data storage in a file format on the cloud involves several steps. First, the sensor data collected by the microcontroller needs to be formatted and processed. Then, using a suitable software or platform such as ThingSpeak, the microcontroller establishes an internet connection and sends the formatted data to the cloud server. This communication typically occurs over protocols like HTTP or MQTT. The cloud platform stores the data in a structured format, often in a file or database, allowing for easy retrieval and analysis.

In the given observation table, sample values for the data stored in the cloud could be:
- Height of Object: 120 cm
- Distance: 80 cm
- Angular Position: 45 degrees
- RGB Color: Blue
- Proximity: Detected
- Force: 10 N
- Angular Position: 90 degrees
- Acceleration: 2 m/s^2
- Angular Velocity: 100 deg/s
- Flexion: 30 degrees

Observation table for data storage – Processed data entry is expected in observation table

Sr. no	Name of Sensor	Parameter sensed	Reading 1	Reading 2	Reading 3	Reading 4	Average Value
1	Depth	Height of Object	125 cm	124 cm	126 cm	123 cm	124.5 cm
2	Distance	Distance	85 cm	84 cm	86 cm	83 cm	84.5 cm
3	Position	Angular Position	46 degrees	47 degrees	45 degrees	48 degrees	46.5 degrees
4	Imaging	RGB Color	Green	Blue	Red	Green	N/A
5	Proximity	Proximity	Detected	Not Detected	Detected	Detected	N/A
6	Force/Torque	Force	11 N	10.5 N	11.2 N	10.8 N	10.9 N
7	Rotation	Angular Position	92 degrees	91 degrees	93 degrees	90 degrees	91.5 degrees
8	Accelerometer	Acceleration	2.5 m/s^2	2.3 m/s^2	2.6 m/s^2	2.4 m/s^2	2.45 m/s^2

9	Inertial Measurement Unit	Angular Velocity	105 deg/s	102 deg/s	106 deg/s	103 deg/s	104 deg/s
10	Flex	Flexion	32 degrees	31 degrees	33 degrees	30 degrees	31.5 degrees

h. Result & Conclusion – Elaborate your learning and findings after completion of Module 2.

Results :

- Identified components for IoT-based automation systems including sensors, actuators, and controllers for various applications such as Smart Agriculture, Smart Home Automation, etc.
- Designed PCB layouts using software tools like Eagle or KiCad and developed PCBs using fabrication methods.
- Connected components to microcontrollers like Arduino or NodeMCU ensuring proper wiring and integration.
- Conducted circuit analysis and testing to verify functionality and perform integration testing.

Conclusion :

This project provided hands-on experience in IoT system implementation, enhancing students' understanding of component selection, circuit design, and testing processes. By successfully completing the project, students gained practical skills applicable to real-world scenarios in agriculture, home automation, smart cities, disaster management, military, and industrial automation.