Interdisciplinary Molecular Communication & Sensing Research Group

Interdisciplinary Molecular Communication & Sensing (IMCS) Research Group

Progress Report I

January 2024 – September 2024

Group Details

Title(s) of the Projects Conducted	▪ Prototyping Platform for Molecular Communications (MC) Research ▪ Towards Plant-to-Plant Communication via Molecular Signals
Principle Investigator	Dr. Asanka P. Sayakkara
Co-Investigators	▪ Mr. Kenneth Thilakarathna ▪ Dr. Randil Pushpananda ▪ Dr. Lasanthi De Silva ▪ Dr. Kasun Karunanayake ▪ Dr. Chamath Keppitiyagama ▪ Dr. Chaminda Ranasinghe
Research Assistant	Ms. S. A. V. S. Piyathilake
Other Members	▪ Oshani E. Wickramasinghe ▪ Vishmina Obeysekara ▪ Gayathri Purage
Reporting Period	01.01.2024 – 30.09.2024
Outcomes/ Deliverables	▪ Initial implementation of a cost-effective research platform for the exploration of MC ▪ Conference Paper - Title of Paper: A Low-Cost, Off-the-Shelf Prototyping Platform for Molecular Communications Research - Title of Conference: 17th International Conference on Signal Processing and Communication Systems (https://icspcs2024.io.pbs.edu.pl/)

Table of Contents

1. Prototyping Platform for Molecular Communications (MC) Research

1.1. Literature Review

Initially, the literature in the domain of molecular communication was reviewed by all team members. The goal was to identify unexplored areas, assess what had already been accomplished, and study various related topics as outlined below.

- Overviews and Tutorials
- Special Issues
- Modeling of Transceiver and Propagation Biophysics
- Performance Evaluation, Resource Management, and Parameter Estimation
- **Simulation, Experiments, and Testbeds**
- Standardization and Emerging Applications
- Books

The referenced literature can be accessed here:

https://github.com/UCSC-IMCS/IMCS-Documentation/tree/master/Literature%20Review

After reviewing the literature and gaining an understanding, it was collectively agreed by the team to focus on the topic of simulation, experimentation, and testbeds, as this was considered the best option for conducting research in molecular communication using computer science knowledge.

In the literature, it was found that researchers have attempted to establish molecular MC between two computers, using chemical molecules as carriers with a custom-built setup acting as the receiver. However, to the best of our knowledge, these studies have relied on a guided flow to transmit chemical signals. Therefore, the research team agreed to implement a setup that establishes MC between two computers using a wireless diffusion mechanism with atmospheric air, without employing any guided media or air flow control mechanisms, relying solely on the chemical properties of diffusion and atmospheric conditions to explore the potential for developing an MC system. It was also decided that the setup should be cost-effective.

1.2. Implementation of the MC Research Platform

1.2.1. Selection of Sensors and Chemicals

Since the initial step focused solely on implementing the receiving end of the molecular MC system, suitable chemicals and sensors capable of transmitting and receiving chemical signals were identified. The literature revealed that several studies utilized ethanol as the chemical and the MQ3 sensor as the detection device. The MQ3 sensor is a volatile organic compound (VOC) sensor. Therefore, it was decided that, as a starting point, ethanol and the MQ3 sensor would be appropriate choices. In addition to the MQ3 sensor, other MQ-series sensors and various additional sensors were also utilized. Furthermore, in later experiments, thinner was selected as the chemical due to its common availability and low cost.

The specific sensors and chemicals employed in this research are listed below.

Sensors	Chemicals
▪ MQ-2 ▪ MQ-214 ▪ MQ-3 ▪ MQ-5 ▪ MQ-6 ▪ MQ-8 ▪ MQ-9 ▪ MICS-4514 Data Sheets: https://github.com/UCSC-IMCS/VOC-Sensors/tree/master/VOC%20Sensors%20Datasheets	▪ 95% Ethanol ▪ N-C Thinner

1.2.2. Individual Sensor Testing

As we were not familiar with the functionality of the sensors, we decided to test each one individually to better understand how they operate.

The MQ series of gas sensors operates by detecting specific gases through a sensitive metal oxide layer that changes resistance in response to gas adsorption.

Each MQ sensor typically features the following pins:

- VCC: Power supply (usually 5V).

- GND: Ground connection.

- Analog Output Pin: Outputs a voltage corresponding to the gas concentration.

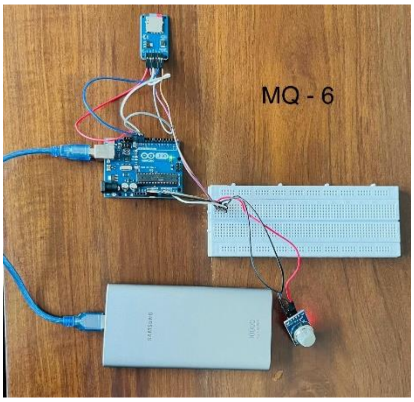
- Digital Output Pin: Provides a binary signal when a specific threshold concentration is exceeded.

Connection Steps:

1. Connect the VCC pin of the MQ sensor to the 5V pin on the Arduino board.

2. Connect the GND pin of the MQ sensor to one of the GND pins on the Arduino board.

3. Connect the Analog Output Pin to an analog input pin on the Arduino (e.g., A0).



The MQ sensors require a preheating period to stabilize before accurate readings can be obtained. After this warm-up phase, they can provide real-time data, allowing for effective monitoring of environmental conditions.

The MICS-4514 sensor operates similarly, using a sensitive metal oxide layer that detects gases through changes in electrical resistance caused by gas molecule interaction.

The MICS-4514 sensor features multiple output pins, including:

- VCC: Power supply.

- GND: Ground connection.

- Analog Output Pins (Vnox & Vred): Provide continuous voltage levels corresponding to the concentrations of the target gases.

- Digital Output Pin: Offer threshold detection.

- Preheat Pin: Should be connected to a digital pin on the Arduino board to control the heating element.

Connection Steps:

1. Connect the VCC pin of the MICS-4514 sensor to the 5V pin on the Arduino board.

2. Connect the GND pin of the MICS-4514 sensor to one of the GND pins on the Arduino board.

3. Connect the Analog Outputs Pins to analog input pins on the Arduino.

5. Connect the Preheat Pin to a digital pin on the Arduino (e.g., D2) to control the heating element.

Throughout our testing, we logged the analog outputs from both the MQ and MICS sensors to an SD card in CSV format. We identified that while logging data to an SD card connected to the Arduino board, we cannot capture real-time data unless we connect a timer to the Arduino for accurate time-stamping.

	A	B	C	D
1	Time	Vnox	Vred	
2	10951	160	774	
3	11976	157	773	
4	12999	154	769	
5	14024	152	766	
6	15047	151	765	
7	16072	151	764	
8	17095	150	762	
9	18120	149	760	

The code related to individual testing can be found here.

https://github.com/UCSC-IMCS/VOC-Sensors/tree/master/Individual%20Sensor%20Testing

1.2.3. Implementation of the Overall System

After testing the sensors individually, we proceeded to test combinations of sensors, such as groups of three and four, connected to the Arduino board.

Since we were unable to log timestamps when using SD card logging, we decided to connect the Arduino board to a laptop/computer, listen to the serial port, and log the sensor readings with timestamps directly on the laptop/computer. To achieve this, we used a Python script for data logging.

	A	B	C	D	E	F	G	H	I	J
1	Timestamp	mq2	mq3	mq5	mq6	mq8	mq9	mics_Vred	mics_Vnox	mq214
2	7/16/2024 13:16	324	95	45	230	35	204	328	969	1012
3	7/16/2024 13:16	318	94	46	230	35	198	458	917	1012
4	7/16/2024 13:16	309	92	46	224	35	192	275	914	1012
5	7/16/2024 13:16	301	91	45	220	35	188	130	911	1012
6	7/16/2024 13:16	296	89	43	216	35	185	94	909	1012
7	7/16/2024 13:16	296	89	43	216	34	186	158	907	1012
8	7/16/2024 13:16	296	89	43	216	34	187	133	906	1012
9	7/16/2024 13:16	291	87	42	212	34	183	98	904	1012
10	7/16/2024 13:16	285	86	41	208	34	178	74	902	1012
11	7/16/2024 13:16	279	86	40	205	33	173	80	901	1012
12	7/16/2024 13:16	274	85	40	203	33	172	62	899	1012
13	7/16/2024 13:16	273	84	39	202	33	171	63	898	1012
14	7/16/2024 13:16	272	83	39	200	33	169	78	896	1012

We encountered a significant challenge when trying to connect nine analog output pins from our sensors to the Arduino board. The standard Arduino Uno only provides six analog input pins, which wasn't enough to handle all the sensors we needed to connect simultaneously. This limitation required us to explore alternative solutions.

Our first approach was to try an ESP32 board, which offers up to 16 ADC (analog-to-digital converter) channels. While this seemed promising due to its higher pin count and more powerful processor, we couldn't get the sensors to function reliably, possibly due to differences in signal handling or voltage requirements between the sensors and the board.

Thus, we decided to switch to the Arduino Mega 2560, which has 16 dedicated analog input pins—more than enough to accommodate all nine sensor connections. This choice provided a straightforward solution, as the Mega's architecture is similar to the Uno, making it easy to integrate into our existing setup while offering the additional inputs we needed.

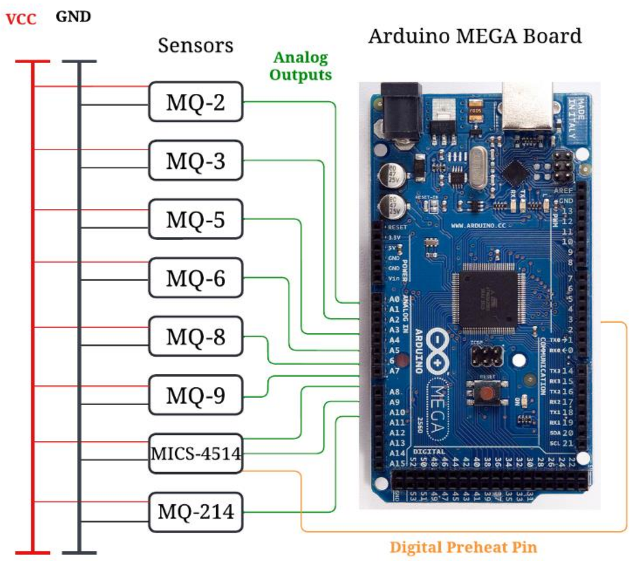
To set up and run this system properly, follow these steps in detail:

1. Sensor and Arduino Mega Connections:

- ✓ First, connect the analog output pins of the sensors to the corresponding analog input pins on the Arduino Mega board. This allows the Arduino to read sensor data.
- ✓ For the MIC sensor, connect its preheat pin to one of the digital input pins on the Arduino Mega board. This digital input will be used to manage the preheating process for the sensor.
- ✓ Next, connect the VCC and GND pins of all sensors to the VCC and GND rails on a breadboard, respectively. Similarly, connect the VCC and GND pins of the Arduino Mega board to the same VCC and GND rails on the breadboard. This setup ensures a common power supply and ground across all components.

2. Uploading Code to the Arduino:

- ✓ After making the hardware connections, connect the Arduino Mega board to your computer or laptop using a USB cable.
- ✓ Open the Arduino IDE on your computer, upload the Arduino code.

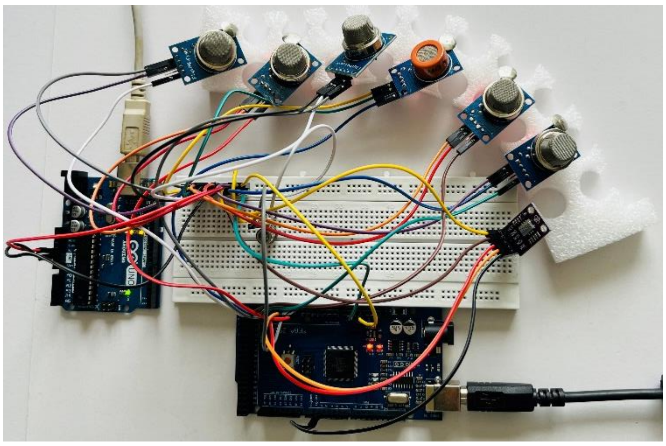


3. Running the Python Script for Data Logging:

- ✓ After successfully uploading the Arduino code, you can start logging the sensor data by running the Python script on your computer.

IMPORTANT

The power provided by the Arduino Mega board alone wasn't sufficient to power all the connected sensors. To resolve this, an additional power source was used. Specifically, the VCC and GND pins from another Arduino board were connected to the same VCC and GND rails on the breadboard, which are shared with the sensors. This additional power source helped supply adequate voltage and current to all the sensors.



You can access the code through the following links:

Arduino code:

https://github.com/UCSC-IMCS/IMCS-VOC_Sensors/tree/master/AllInOne%20-%20Arduino%20MEGA/arduino%20code

Python script:

https://github.com/UCSC-IMCS/IMCS-VOC_Sensors/blob/master/AllInOne%20-%20Arduino%20MEGA/serial%20port%20listen.ipynb

---

The current sensor setup is bulky and lacks portability. As part of future work, a dedicated device has been developed to house the sensor system, allowing it to operate independently without the need for connection to a computer or laptop. Further details about this setup will be provided in the next progress document.

1.3. Experimentation & Testing

We initially used ethanol in our experiments and conducted three types of tests as follows:

- **Ethanol Sensitivity Test:** The first experiments were carried out to determine if the sensors were sensitive to ethanol.
- **Distance Variation Experiments:** After confirming the sensors' sensitivity, we conducted tests by placing ethanol at different distances from the sensor setup.
- **Timed Exposure Experiments:** In this stage, we exposed the sensor setup to ethanol at fixed time intervals, such as every hour, at a constant distance, then removed the ethanol and repeated the process.

As the next step, since ethanol is relatively expensive and not commonly used in everyday applications, we switched to thinner.

- Using thinner, we shortened the time intervals to 15 minutes and conducted timed exposure experiments at a fixed distance.

The summary of the experiments is available here.

- Ethanol: https://github.com/UCSC-IMCS/IMCS-Documentation/blob/master/Experiments%20and%20Results/Summary%20of%20the%20Ethanol%20Experiments.pdf
- Thinner: https://github.com/UCSC-IMCS/IMCS-Documentation/blob/master/Experiments%20and%20Results/Summary%20of%20the%20Thinner%20Experiments.pdf

1.4. Deliverables

Based on the completed experiments and the results obtained, a conference paper has been prepared and submitted to the 17th International Conference on Signal Processing and Communication Systems, where it has been accepted.

You can access the paper and supporting materials through this link.

https://github.com/UCSC-IMCS/IMCS-Documentation/tree/master/Conference%20Papers/ICSPCS%20Paper%20accepted

2. Towards Plant-to-Plant Communication via Molecular Signals

2.1. Growing Plants

In parallel with developing a rapid prototyping platform for molecular communications research, we explored the possibility of studying plant-to-plant communication. This research is driven by the potential future application of molecular communication mechanisms for eco-friendly pest control, providing an alternative to harmful chemical pesticides.

Alongside the implementation of the sensor setup, we attempted to grow plants indoors to test interactions using the sensor system.

The main plant species we focused on were

- Tomato
- Basil

However, the indoor growing conditions were not successful, which impacted our ability to conduct the experiments as planned.

As an opportunity to enhance our research, we collaborated with the Plant Science Department at the university, which provided access to a SmartPot designed for growing plants in indoor settings. We utilized this innovative system to cultivate mint plants.

2.2. Experimentation & Testing

Using the SmartPot along with the implemented sensor setup, we conducted a series of experiments to investigate plant responses. Initially, we placed the sensor setup inside the SmartPot and recorded baseline readings from the sensors. After establishing these baseline measurements, we plucked some leaves from the mint plants and recorded the sensor readings again to determine if there were any changes in response to this disturbance.

For more detailed information about the experiments and their findings, you can access the following link.

https://github.com/UCSC-IMCS/IMCS-Documentation/blob/master/Experiments%20and%20Results/Summary%20of%20the%20Mint%20Plants%20Experiments.pdf