Project Brief for ENG20009

Name: Md Redwan Ahmed Zawad

Course: *Bachelor of Engineering(Software)*

Institute: *Swinburne University of Technology*

Location: Melbourne, Australia

Contact:
103501849@student.swin.edu.au

Abstract— **This report describes the design and implementation for developing an SDI-12 Sensor and data logger. Two sensors will be used for building the SDI-12 sensor BME680 and BH1750FVI. Every sensor will be linked to the Arduino using the I2C communication protocol. Upon data reception, the collected information will be consolidated and then transmitted to the UART interface using SDI commands functionality. The primary goals for building this sensor are reading relevant data, programming Arduino based on SDI-12 communication protocol, creating accessibility to select different sensors, saving sensor data, and displaying the data.**

Keywords— ASCII format, microprocessor, environmental monitoring

I. INTRODUCTION (*HEADING 1*)

SDI-12, a standard communication protocol, serves as a medium for transferring measurements from an intelligent sensor to a data recorder. It makes logical sense to connect microprocessor-based sensors to a data recorder using a serial-digital interface. Both sensors and data recorders can gain significantly from this strategy. The self-calibration algorithms used by microprocessor-based sensors can be complex and one-of-a-kind. The flexibility to swap out sensors without having to update the data recorder with calibration or other information is a notable advantage. In addition, the interface provides power to the sensors as a secondary purpose. Compact sensor units can now combine a microprocessor, power supply regulation, and other necessary hardware thanks to the use of hybrid circuitry and surface mount technology [1]. For its usefulness, many companies like Thermo Fisher Scientific Australia Pty Ltd use it for their hardware and software services.

To build an SDI-12 sensor, two raw sensors- BME680 and BH1750FVI will be used which have been provided by Swinburne Institute of Technology as part of the Engineering Technology Inquiry project. To create a thorough understanding of the SDI-12 protocol, an application of the SDI-12 has been studied intently. Combined with the hands-on experience with Arduino, it will help the participants to develop the practical and theoretical knowledge required for engineering technology difficulties. Through establishing a deep understanding of Arduino, this unit aims to enable students to experience the complex relationship between hardware and software.

This SDI-12 sensor will be able to read the relevant data from each sensor. It will be given the capability to choose a sensor to be used from a menu. It will also be able to save the selected data and RTC into the SD card while displaying the data on LCD with relevant graphics. The sensors have some limitations. For instance, the BME680 sensor has a limited measurement range and requires a warm-up time for accurate gas resistance measurement, calibration requirement, sensitivity to contaminants, etc. While BH1750 has some common limitations as BME680, it has constraints of accuracy at extreme conditions, spectral sensitivity, ambient light influence, etc.

II. COMPONENTS

BME680

Fig 1. BME680

The BME680 is a ground-breaking gas sensor that combines very accurate pressure, humidity, and temperature sensing functions with precise and linear gas sensing capabilities. It is carefully constructed to meet the needs of wearable technology and mobile applications, with a focus on small size and low power consumption. The BME680 ensures favorable power usage, ongoing dependability, and pronounced resistance to electromagnetic interference depending on the operational mode. The gas sensing component in BME680 can identify a wide variety of gases, including volatile organic compounds (VOCs), in the context of evaluating personal health-related air quality [2].

BH1750FVI

The BH1750 is an ambient light sensor integrated circuit that uses the I2C bus interface and operates in the digital domain. Through the I2C interface, this module manages difficult duties and provides a simple digital output that displays a number indicating brightness in Lux [3]. Adjusting the brightness of the display in mobiles and LCDs is the primary use of this sensor.

Fig. 2. BH1750FVI

I2C

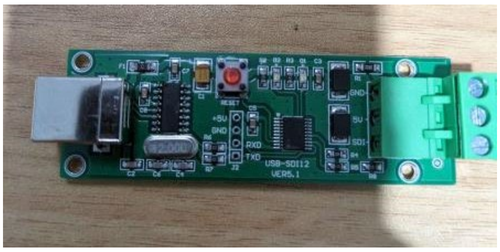


Fig. 3. I2C

Due to the ease of integration of I2C into different electronic designs, which requires communication between a primary master device and numerous secondary slave devices, or even between multiple master devices, the I2C communication bus has grown significantly in popularity and has been widely adopted across diverse electronic devices. The simple implementation is related to the requirement of just two wires for communication facilitation. With support for up to almost 128 devices with 7-bit addressing and up to nearly 1024 devices with 10-bit addressing, this configuration provides seamless interaction between a wide range of devices [6].

SDI-12

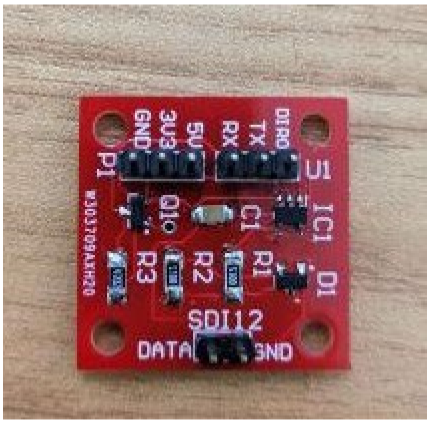


Fig. 4 SDI-12

Serial Data Interface at 1200 Baud, sometimes known as SDI-12, is a standardized communication protocol. This protocol makes it easier to transfer measurement data from an environmental data acquisition (EDA) sensor driven by a microprocessor to a battery-operated data logger. The EDA sensor takes measurements, uses the raw data to conduct the necessary computations, and then displays the results in terms that are simple to comprehend. For instance, a pressure sensor records several pressure readings, averages them, and then reports the pressure information in psi, inches of mercury, bars, millibars, or torrs. These calculations are handled by the sensor's internal CPU, which also transforms sensor values into the proper units and uses the SDI-12 protocol to send measurements to the data logger [3].

III. Application Of SDI-12

One of the main applications of SDI-12 is research. It is being used in building water quality measurement systems, measuring soil moisture, developing a mote for horticulture, and many more. One such research study involving hardware and software will be discussed here.

Development of a Raspberry Pi-based, SDI-12 sensor environmental data logger

Based on the Raspberry Pi, developing a low-cost and reliable data logger was the aim of the project which would benefit the environment, businesses, individuals, and communities. Configuration of SDI-12 sensors, recording measurements, obtaining results, and saving data were handled by the software interface. The software was simple enough to be understood and modified by others. Only basic knowledge of computers, access to Raspberry Pi, and assembly skills would be required to set up the system [5].

The components and hardware used for the projects were Raspberry Pi, SDI-12 sensors, wiring and power supply, data storage, and RTC. C++ was used to create software for data logger and SDI-12 library was used for this purpose. The code was written to handle the following tasks: collecting data, storing the data, implementing error handling, and validating data. The developed Software was made open source. Raspberry Pi was chosen over Arduino because of the inability of Arduino to perform complex programming and limited memory and processing power [5].

UART and GPIO were considered for the implementation of SDI-12. In the end, GPIO was favored over UART as some of the designs regarding UART were vague and uncertain of successful outcomes. SDI-12 library for Arduino-based data logger was modified to support the Raspberry Pi-based data logger. Some of the functions were modified while some new functions were added to the library. The public functions that were inherited from the SDI-12 library would be executed in the following sequence: begin(),sendCommand(cmd), end(). begin() would start the communication, sendCommand(cmd) was used to send commands to the sensor, and end() signified the completion of the communication exchange [5].

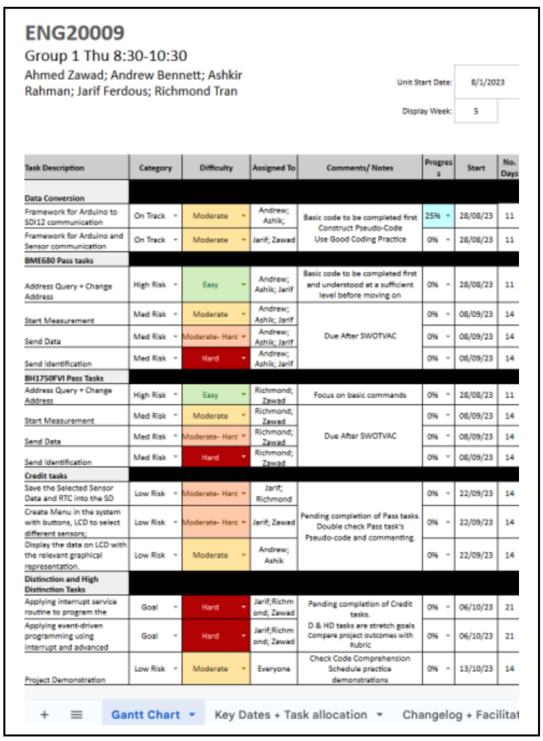
In the first step of the implementation of the Raspberry Pi library, a command is sent to the sensor. After a certain period, the end() function would disable the interrupt. Depending on the one response, the buffer would be flushed, and the command would be resent if the error status was true. If the error status returns false, the buffer will be read before flushing and proceeding to the next command. Key functions were stored in a text file to allow the logger to collect data and store values in a configuration file. When a new sensor is installed or removed the configuration file would be changed. The main function, measurement handling functions, SD-12 device configuration functions, and generic functions are the main parts of the SDI-12 logger program. A user could interact with the data logger through the command prompt [5].

The Raspberry Pi was configured to interface with the SDI-12 sensors to enable continuous data flow. The Raspberry Pi had to organize the data, measure the sensors' output, and control the information flow. This data was stored locally on the storage device of the Raspberry Pi or transferred to distant servers for further processing and archiving [5].
The data logger was tested by sending an address query 30 times to a sensor and correct answers were returned. The data logger was tested using two Decagon GS3 sensors and one Decagon 5TM sensor. 880 lines of data were recorded within 75 hours where data was recorded every 5 minutes. 63 erroneous values and diagnostic errors were found in the file [5].
In conclusion, it was clear that an Arduino library could be used with the Raspberry Pi-based logger. An oscilloscope was used to design and test the hardware. Even though Raspberry Pi and the SDI-12 are reliable, the rate of corrupt data in the data file was quite high. GPIO method should be studied further to assess its suitability. Experience with using the Raspberry Pi command line is required for HMI. The command prompt must be used for program execution and HMI navigation. Raspberry Pi data logger can be used if 4 watts of power can be provided continuously and protected from the environment. The Raspberry Pi can run continuously for a long time [5].

IV. PROJECT OVERVIEW AND TASK DIVISION

A group of five students will be collaborating to develop the SDI-12 sensor and Data logger. General tasks involve planning, design, construction, programming, testing, and debugging. After completion of the project, it will be tested using UART to SDI-12 and SDI-12 to UART converter. The tasks have been divided between the team members after going through various discussions and meetings. Task Division:

1. Zawad: Framework for Arduino and Sensor communication. Address Query and Change Address, Start Measurement, Send Data, Send Identification for BH1750FVI. Create a Menu in the system with buttons, and LCD to select different sensors, apply interrupt service routine to program the SDI-12 sensor, apply event-driven programming using interrupt and advanced interrupt-driven programming, and project demonstration.
2. Jarif: Framework for Arduino and Sensor communication. Address Query and Change Address, Start Measurement, Send Data, Send Identification for BME680. Save the selected sensor data and RTC into the SD. Create a menu in the system with buttons and LCD to select different sensors. Applying interrupt service routine to program the SDI-12 sensor, applying event-driven programming using interrupt and advanced interrupt-driven programming and project demonstration.
3. Richmond (team Coordinator): Address Query and Change Address, Start Measurement, Send Data, Send Identification for BH1750FVI. Save the selected sensor data and RTC into the SD. Applying interrupt service routine to program the SDI-12 sensor, applying event-driven programming using interrupt and advanced interrupt-driven programming and project demonstration.
4. Andrew: Framework for Arduino to SDI-12 communication, Address Query and Change Address, Start Measurement, Send Data, Send Identification for BH1750FVI. Display the data on LCD with the relevant graphical representation and project demonstration.
5. Ashik: Framework for Arduino to SDI-12 communication, Address Query and Change Address, Start Measurement, Send Data, Send Identification for BH1750FVI. Display the data on LCD with the relevant graphical representation and project demonstration.


ENG20009
Group 1 Thu 8:30-10:30
Ahmed Zawad; Andrew Bennett; Ashkir Rahman; Jarif Ferdous; Richmond Tran

Unit Start Date: 8/1/2023
Display Week: 5

Task Description	Category	Difficulty	Assigned To	Comments/ Notes	Progress	Start	No. Days
Data Conversion							
Framework for Arduino to SDI12 communication	On Track	Moderate	Andrew; Ashik;	Basic code to be completed first Construct Pseudo-Code Use Good Coding Practice	25%	28/08/23	11
Framework for Arduino and Sensor communication	On Track	Moderate	Jarif; Zawad		0%	28/08/23	11
BME680 Pass tasks							
Address Query + Change Address	High Risk	Easy	Andrew; Ashik; Jarif	Basic code to be completed first and understood at a sufficient level before moving on	0%	28/08/23	11
Start Measurement	Med Risk	Moderate	Andrew; Ashik; Jarif	Due After SWOTVAC	0%	08/09/23	14
Send Data	Med Risk	Moderate- Hard	Andrew; Ashik; Jarif		0%	08/09/23	14
Send Identification	Med Risk	Hard	Andrew; Ashik; Jarif		0%	08/09/23	14
BH1750FVI Pass Tasks							
Address Query + Change Address	High Risk	Easy	Richmond; Zawad	Focus on basic commands	0%	28/08/23	11
Start Measurement	Med Risk	Moderate	Richmond; Zawad	Due After SWOTVAC	0%	08/09/23	14
Send Data	Med Risk	Moderate- Hard	Richmond; Zawad		0%	08/09/23	14
Send Identification	Med Risk	Hard	Richmond; Zawad		0%	08/09/23	14
Credit tasks							
Save the Selected Sensor Data and RTC into the SD	Low Risk	Moderate- Hard	Jarif; Richmond	Pending completion of Pass tasks. Double check Pass task's Pseudo-code and commenting.	0%	22/09/23	14
Create Menu in the system with buttons, LCD to select different sensors;	Low Risk	Moderate- Hard	Jarif; Zawad		0%	22/09/23	14
Display the data on LCD with the relevant graphical representation.	Low Risk	Moderate	Andrew; Ashik		0%	22/09/23	14
Distinction and High Distinction Tasks							
Applying interrupt service routine to program the	Goal	Hard	Jarif;Richmond; Zawad	Pending completion of Credit tasks. D & HD tasks are stretch goals Compare project outcomes with Rubric	0%	06/10/23	21
Applying event-driven programming using interrupt and advanced	Goal	Hard	Jarif;Richmond; Zawad		0%	06/10/23	21
Project Demonstration	Low Risk	Moderate	Everyone	Check Code Comprehension Schedule practice demonstrations	0%	13/10/23	14

+ Gantt Chart | Key Dates + Task allocation | Changelog + Facilitat

Fig. 5. Task Division

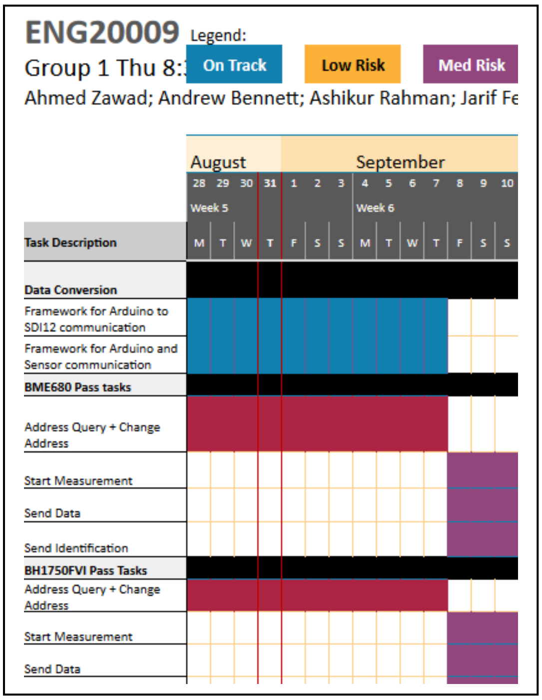


Fig. 6. Time Allocations

V. Task Implementation

To make Arduino understand the SDI-12 communication protocol first we need to connect the SDI-12 to the Arduino which will be connected to the PC and programmed via Arduino IDE. With the help of existing SDI-12 libraries, communication with SDI-12 communication can be simplified. After installing and including the SDI-12 library in the code, an SDI-12 object will be created and SDI-12 communication will be initialized in the setup. Using 'sendCommand()' and 'readData()' functions SDI-12 commands can be provided and data can be read from the sensor.

To create a menu in the Arduino buttons and LCD will be connected to the Arduino using jumpers. Then LiquidCrystal library will be installed into the IDE and included in the code. A LiquidCrystal object will be initialized. Using the necessary logic and code, the menu to select the sensors will be created.

VI. Conclusion

Undoubtedly, SDI-12 sensors are one of the most important tools for various fields like research, environment data recording, etc. It proves to be significant in the world of electronics whose importance cannot be emphasized with words. Developing an SDI-12 will enable the participants of this project to research, learn, and solve problems. A lot of knowledge has been acquired for this project. However, there is a gap of knowledge regarding interrupt-based programming and advanced interrupt-driven programming which the team members are trying to fill through teamwork and various sources. The team realizes that this project will be most advantageous for the future and continues to work determinately to achieve working SDI-12 sensors and data logger.

References

[1] www.sdi-12.org. (n.d.). SDI-12 Support Group. [online] Available at: https://www.sdi-12.org/#:~:text=SDI%2D12%20is%20a%20standard [Accessed 31 Aug. 2023]

[2] Bosch Sensortec. (n.d.). BME680. [online] Available at: https://www.bosch-sensortec.com/products/environmental-sensors/gas-sensors/bme680/.

[3] core-electronics.com.au. (n.d.). BH1750FVI Digital Light Intensity Sensor Module (GY-302). [online] Available at: https://core-electronics.com.au/bh1750fvi-digital-light-intensity-sensor-module-gy-302.html.

[4] SDI-12: EVERYTHING YOU NEED TO KNOW TO BE SUCCESSFUL IS THE RESEARCHER'S COMPLETE GUIDE. (n.d.). Available at: https://publications.metergroup.com/Sales%20and%20Support/METER%20Environment/Website%20Articles/sdi-12-everything-need-know-successful.pdf.

[5] Coppock, J. (2015). Development of a Raspberry Pi based, SDI-12 sensor environmental data logger. [online] sear.unisq.edu.au. Available at: https://sear.unisq.edu.au/29188/ [Accessed 1 Sep. 2023].

[6] HowToMechatronics (2015). How I2C Communication Works & How To Use It with Arduino. [online] HowToMechatronics. Available at: https://howtomechatronics.com/tutorials/arduino/how-i2c-communication-works-and-how-to-use-it-with-arduino/.