Conceptual implementation of a data logger with a graphical user interface data extraction program

Ahmad Shukri Fazil Rahman, Samila Mat Zali, Shamshul Bahar Yaakob, Abdul Rahim Abdul Razak, Mohd Zamri Hasan, Siti Hajar Mohd Tahar
Pusat Pengajian Kejuruteraan Sistem Elektrik, Universiti Malaysia Perlis, Kampus Alam UniMAP Pauh Putra, Perlis, Malaysia

Article Info

Article history:

Received Oct 5, 2018
Revised Nov 27, 2018
Accepted Dec 15, 2018

Keywords:

Data logger
Firmware
Microcontroller
Simulation

ABSTRACT

A Graphical User Interface (GUI) design was developed to extract data from a data logger. The low-cost data logger was developed using a common pic18f4550 microcontroller. Simple data recording feature is implemented by using the Electrically Erasable Programmable Read-Only Memory (EEPROM) module from the microcontroller unit. The GUI software employed VB.net 2013 and has been tested to work in window 10 environment. Simulation work was performed with temperature and humidity sensors using Proteus VSM software. Upon simulation, the GUI shows its capability of extracting measured data from the data logger and converting the data to a Comma-Separated Values, (CSV) format for further analysis.

Corresponding Author:

Ahmad Shukri Fazil Rahman,
Pusat Pengajian Kejuruteraan Sistem Elektrik,
Universiti Malaysia Perlis,
Kampus Alam UniMAP Pauh Putra, 02600 Arau,
Perlis, Malaysia.
Email: ahmadshukri@unimap.my

1. INTRODUCTION

Data monitoring provided an important information for a comprehensive, accurate picture of the conditions being monitored, such as air temperature, humidity, voltage, current, pressure etc. [1, 2]. Common monitoring techniques employing a data logger provides a link between the measured sample and the collected data. Upon activation, data loggers can be left unattended to perform measurement for the duration of the monitoring period.

Early logging systems include data acquisition board and its associated computer software utility enabled user to monitor conditions[3]. As technology advances, a new generation of digital systems called microcontrollers arise[4, 5]. These microcontrollers are low cost smart embedded systems that can be designed and assembled easily.

Recorded data from this logger can be manipulated provided that easy access and data retrieval can be taken making the monitored data suitable for analysis[6]. This implies that reliable connection exists such as digital interfaces like Inter Integrated Circuit (I2C), *Serial Peripheral Interface* (SPI) or Universal Asynchronous Receiver/Transmitter (UART) enabling the direct interconnection of a digital sensor, data logger and a computer.

Hence this article presents a complement application between a data-logger and a retrieval application program. This article highlights the configuration of a data logger system which includes a microcontroller and two sensors. Simulation work is carried out to justify the operation of the data logger with its results.

2. PROPOSED SYSTEM IMPLEMENTATION

Data logging was performed by using a PIC18F4550 microcontroller unit from Microchip. Through EEPROM module, recorded data will be stored to the module. Figure 1 shows the firmware flowchart for the proposed system. Three inputs are available, i.e. reset EEPROM, Logging and Transmission.

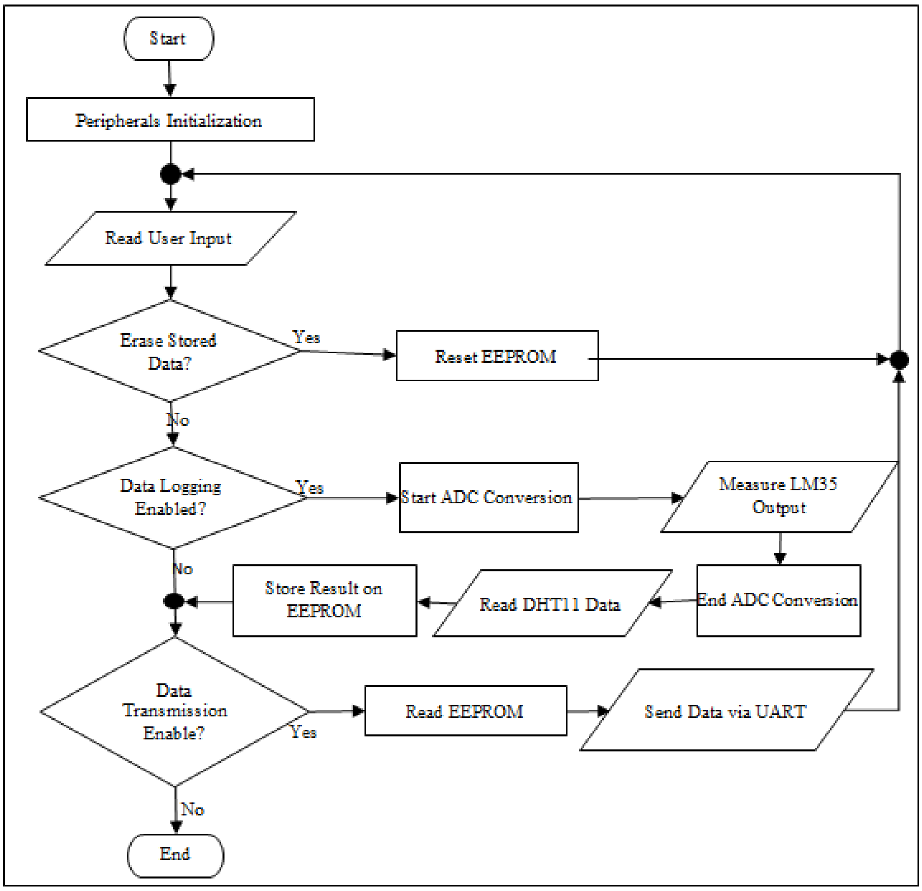


Figure 1. Firmware Flowchart for Data Logger System

The program begins by reading input from user. In data logging mode, input from sensors will be sampled every 3 second. The Analog to Digital Converter (ADC) will start reading the temperature. The ADC is a 10-bit resolution ranging from 0 to 1023 step size, the conversion process is completed by implementing Equation 1.

$$ADC\ conversion = \frac{Temperature \times 0.010\frac{V}{°C}}{5V} \times 1023 \qquad (1)$$

The converted value will be stored in EEPROM. The 10-bit data from ADC will be converted to an 8-bit data before being stored in EEPROM. Successive operation will be for humidity sensor. Serial input from humidity sensor output will be read and stored into EEPROM.

The reset function will perform EEPROM erased, while data transmission process enables recorded data to be sent via UART to a computer/laptop. Figure 2 shows the program flow for data retrieval routine (GUI).

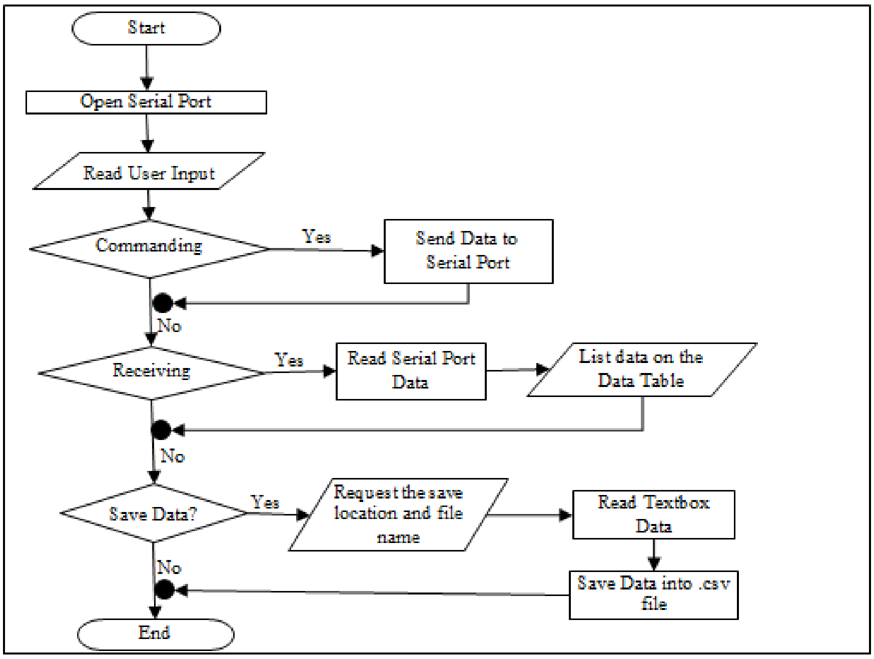


Figure 2. GUI Flowchart for Data Retrieval

Visual Studio 2013 is used to develop the GUI that could accommodate data transmission between the data logger and a computer[7]. Upon port selection, the program will commence data logging reading from data logger. Available data will be displayed on the screen and can be saved in CSV format[8, 9].

3. SYSTEM SIMULATION

System simulation was conducted through Proteus VSM. Data logger schematic is shown in Figure 3.

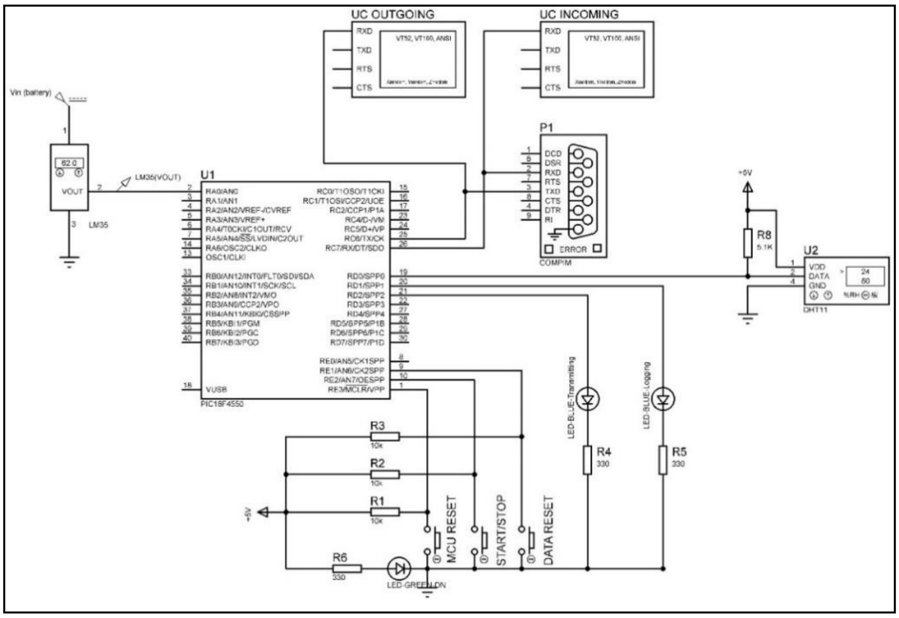


Figure 3. Proposed System Schematic

Data transmission will be transmitted through COMPIM (P1). The temperature sensor is connected to ADC channel port RA0, the humidity sensor is connected to pin RD0 and three switches (START/STOP, DATA RESET and MCU RST) are connected to pin RE1 to RE3 respectively. Two LEDs D2 and D3 (pin RD1 and RD2) will display current progress of system operation. The GUI design is shown in Figure 4.

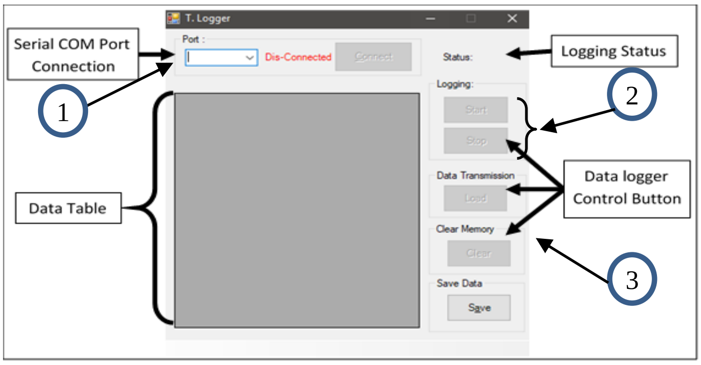
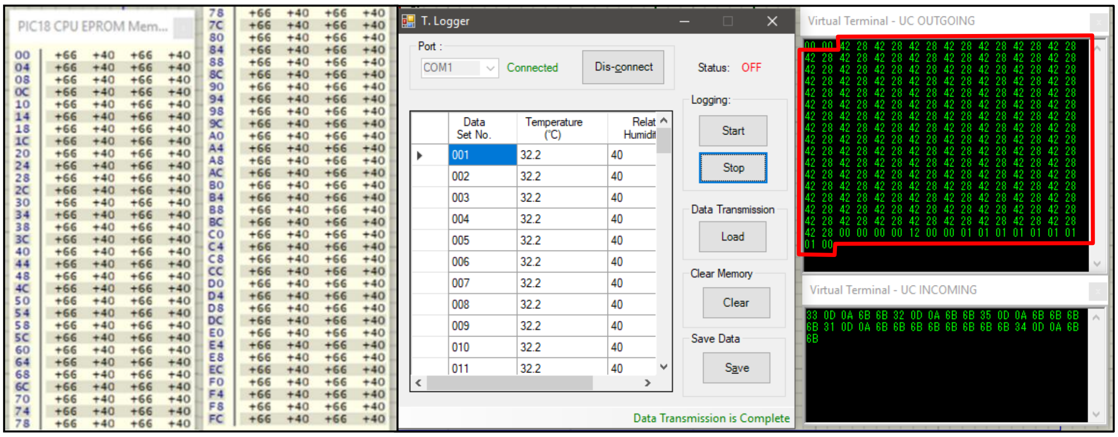


Figure 4. GUI Program

Port selection is available through point 1. System operation can be selected at point 2. Received data will be displayed at Data Table through point 3, while the Save button will record the data in CSV format.

4. RESULTS AND ANALYSIS

Figure 5 shows the overall result from data logger and the GUI. Recorded data from EEPROM is shown in Figure 5a through PIC CPU EEPROM memory map and recorded data will be sent to GUI as shown in Figure 5b.



PIC18 CPU EPROM Mem...

```
00 | +66  +40  +66  +40        78 | +66  +40  +66  +40
04 | +66  +40  +66  +40        7C | +66  +40  +66  +40
08 | +66  +40  +66  +40        80 | +66  +40  +66  +40
0C | +66  +40  +66  +40        84 | +66  +40  +66  +40
10 | +66  +40  +66  +40        88 | +66  +40  +66  +40
14 | +66  +40  +66  +40        8C | +66  +40  +66  +40
18 | +66  +40  +66  +40        90 | +66  +40  +66  +40
1C | +66  +40  +66  +40        94 | +66  +40  +66  +40
20 | +66  +40  +66  +40        98 | +66  +40  +66  +40
24 | +66  +40  +66  +40        9C | +66  +40  +66  +40
28 | +66  +40  +66  +40        A0 | +66  +40  +66  +40
2C | +66  +40  +66  +40        A4 | +66  +40  +66  +40
30 | +66  +40  +66  +40        A8 | +66  +40  +66  +40
34 | +66  +40  +66  +40        AC | +66  +40  +66  +40
38 | +66  +40  +66  +40        B0 | +66  +40  +66  +40
3C | +66  +40  +66  +40        B4 | +66  +40  +66  +40
40 | +66  +40  +66  +40        B8 | +66  +40  +66  +40
44 | +66  +40  +66  +40        BC | +66  +40  +66  +40
48 | +66  +40  +66  +40        C0 | +66  +40  +66  +40
4C | +66  +40  +66  +40        C4 | +66  +40  +66  +40
50 | +66  +40  +66  +40        C8 | +66  +40  +66  +40
54 | +66  +40  +66  +40        CC | +66  +40  +66  +40
58 | +66  +40  +66  +40        D0 | +66  +40  +66  +40
5C | +66  +40  +66  +40        D4 | +66  +40  +66  +40
60 | +66  +40  +66  +40        D8 | +66  +40  +66  +40
64 | +66  +40  +66  +40        DC | +66  +40  +66  +40
68 | +66  +40  +66  +40        E0 | +66  +40  +66  +40
6C | +66  +40  +66  +40        E4 | +66  +40  +66  +40
70 | +66  +40  +66  +40        E8 | +66  +40  +66  +40
74 | +66  +40  +66  +40        EC | +66  +40  +66  +40
78 | +66  +40  +66  +40        F0 | +66  +40  +66  +40
                               F4 | +66  +40  +66  +40
                               F8 | +66  +40  +66  +40
                               FC | +66  +40  +66  +40
```

T. Logger

Port : COM1 Connected Dis-connect

Status: OFF

	Data Set No.	Temperature (°C)	Relat Humidit
▸	001	32.2	40
	002	32.2	40
	003	32.2	40
	004	32.2	40
	005	32.2	40
	006	32.2	40
	007	32.2	40
	008	32.2	40
	009	32.2	40
	010	32.2	40
	011	32.2	40

Data Transmission is Complete

Virtual Terminal - UC OUTGOING

```
00 00 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 42 28 42 28 42 28 42 28 42 28 42 28 42 28
42 28 00 00 00 00 12 00 00 01 01 01 01 01 01 01
01 00
```

Virtual Terminal - UC INCOMING

```
33 0D 0A 6B 6B 32 0D 0A 6B 6B 35 0D 0A 6B 6B 6B
6B 31 0D 0A 6B 6B 6B 6B 6B 6B 6B 6B 34 0D 0A 6B
6B
```

a. Test Data for Transmission Simulation Test

b. Transmission Simulation Test Result

c. Transmission Through Virtual Terminal

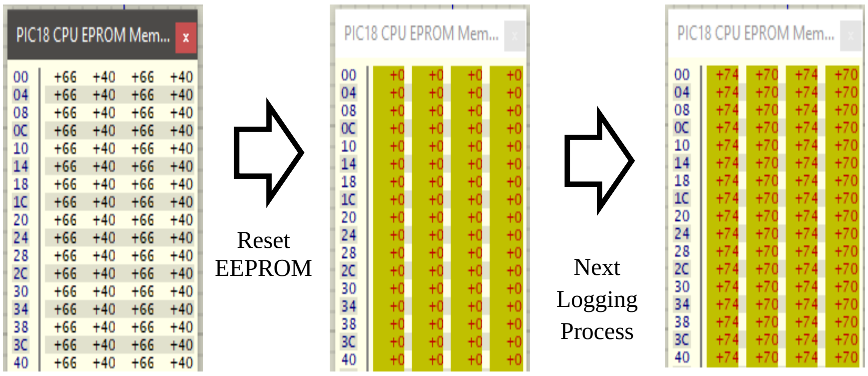
Figure 5. System Output

Two test data from Figure 5a (both 66dec and 40dec) are coming from two sensors, i.e. temperature and humidity sensor. Examination from GUI (Figure 5b) shows that the corresponding values are in match with the original data. These values were visible through virtual terminal (UC OUTGOING in Figure 5c). The region outlined with red color is the test data send by the data logger to the GUI. Initial handshaking protocol are required for both parties to facilitate data transfer. Table 1, summarize the protocol codes involved for a successful transmission. Referring back to Figure 5c, initial code (00Hex) accompanied logged data prior to being submitted. Upon completion, data reset was initiated followed by data re-logging.

Table 1. Protocol Codes for Data Logger Transmission

Control Signal, (Hexadecimal)	Definition	From	To
31	Initiate Data Logging Request	GUI	Data Logger
32	Data Transmission Request		
33	Stop Data Transmission Request		
34	Stop Data Logging Request		
35	Reset Data Memory Request		
00	Not in logging Operation	Data Logger	GUI
01	In logging Operation		
12	Data Reset Memory Completed		

Figure 6 shows memory reset implementation process. Data stored in the EEPROM Module is successfully replaced by zero value. Then, the EEPROM module is tested with another logging operation to measure a simulated temperature of 36°C with percentage of relative humidity of 70%. The digital value “74” and “70” is stored into the EEPROM module during the logging operation.



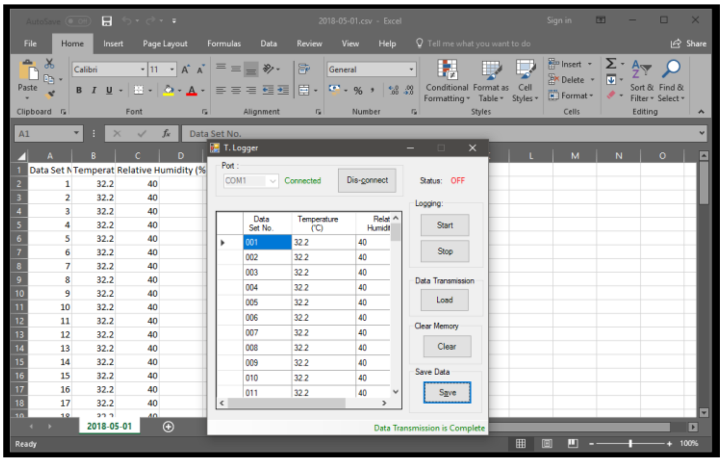
```
00  +66 +40 +66 +40
04  +66 +40 +66 +40
08  +66 +40 +66 +40
0C  +66 +40 +66 +40
10  +66 +40 +66 +40
14  +66 +40 +66 +40
18  +66 +40 +66 +40
1C  +66 +40 +66 +40
20  +66 +40 +66 +40
24  +66 +40 +66 +40
28  +66 +40 +66 +40
2C  +66 +40 +66 +40
30  +66 +40 +66 +40
34  +66 +40 +66 +40
38  +66 +40 +66 +40
3C  +66 +40 +66 +40
40  +66 +40 +66 +40
```

```
00  +0 +0 +0 +0
04  +0 +0 +0 +0
08  +0 +0 +0 +0
0C  +0 +0 +0 +0
10  +0 +0 +0 +0
14  +0 +0 +0 +0
18  +0 +0 +0 +0
1C  +0 +0 +0 +0
20  +0 +0 +0 +0
24  +0 +0 +0 +0
28  +0 +0 +0 +0
2C  +0 +0 +0 +0
30  +0 +0 +0 +0
34  +0 +0 +0 +0
38  +0 +0 +0 +0
3C  +0 +0 +0 +0
40  +0 +0 +0 +0
```

```
00  +74 +70 +74 +70
04  +74 +70 +74 +70
08  +74 +70 +74 +70
0C  +74 +70 +74 +70
10  +74 +70 +74 +70
14  +74 +70 +74 +70
18  +74 +70 +74 +70
1C  +74 +70 +74 +70
20  +74 +70 +74 +70
24  +74 +70 +74 +70
28  +74 +70 +74 +70
2C  +74 +70 +74 +70
30  +74 +70 +74 +70
34  +74 +70 +74 +70
38  +74 +70 +74 +70
3C  +74 +70 +74 +70
40  +74 +70 +74 +70
```

Figure 6. Memory Reset Response

Further simulation is continued for the developed GUI. Through saving feature, the extracted data are saved into the computer is in CSV format. In the simulation, 256 points of measured data are transferred to the GUI and saved into the desired format. The GUI Program Conversion to Excel shown in Figure 7.



Data Set No.

	A	B	C	D
1	Data Set N	Temperat	Relative Humidity (%	
2	1	32.2	40	
3	2	32.2	40	
4	3	32.2	40	
5	4	32.2	40	
6	5	32.2	40	
7	6	32.2	40	
8	7	32.2	40	
9	8	32.2	40	
10	9	32.2	40	
11	10	32.2	40	
12	11	32.2	40	
13	12	32.2	40	
14	13	32.2	40	
15	14	32.2	40	
16	15	32.2	40	
17	16	32.2	40	
18	17	32.2	40	

Port : COM1 Connected Dis-connect

Status: OFF

Logging:

Data Transmission

Clear Memory

Save Data

Data Set No.	Temperature (°C)	Relat Humidi
001	32.2	40
002	32.2	40
003	32.2	40
004	32.2	40
005	32.2	40
006	32.2	40
007	32.2	40
008	32.2	40
009	32.2	40
010	32.2	40
011	32.2	40

Data Transmission is Complete

Figure 7. GUI Program Conversion to Excel

5. CONCLUSION

A microcontroller based data logger system was successfully designed and simulated. In addition, a data retrieval GUI was constructed to complement the complete system. The data logger is equipped with two sensors, i.e. temperature and humidity sensor. Data from sensors were recorded and stored in the EEPROM module. Succeeding process of data retrieval is enabled, data transmission and transformation through GUI was accomplished.

The system capabilities could be improved with external memory, i.e. Secure Digital (SD) card. The additional memory arrangements could prolong the sampling duration. Further improvement such as a real-time clock (RTC) unit to be assembled with existing unit thus provides accurate time logging activities.

ACKNOWLEDGEMENTS

The author would like to acknowledge the support from the Fundamental Research Grant Scheme (FRGS) under a grant number of FRGS/1/2017/TK10/UNIMAP/02/10 from the Ministry of Higher Education Malaysia.

REFERENCES

[1] Mahendra O, Syamsi D, Ramdan A, Astrid M. *Design and implementation of data storage system using USB flash drive in a microcontroller based data logger*. International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT). Bandung, Indonesia. 2015; 58-62.

[2] Fuentes M, Vivar M, Burgos JM, Aguilera J, Vacas JA. Design of an accurate, low-cost autonomous data logger for PV system monitoring using Arduino™ that complies with IEC standards. *Solar Energy Materials and Solar Cells*. 2014;130:529-43.

[3] Engin M. *Open source embedded data logger design for PV system monitoring*. 6th Mediterranean Conference on Embedded Computing (MECO). Bar, Montenegro. 2017; 1-5.

[4] Pasquali V, D'Alessandro G, Gualtieri R, Leccese F. A new data logger based on Raspberry-Pi for Arctic Notostraca locomotion investigations. *Measurement*. 2017;110:249-56.

[5] Carre A, Williamson T. Design and validation of a low cost indoor environment quality data logger. *Energy and Buildings*. 2018;158:1751-61.

[6] Iskandar HR, Purwadi A, Rizqiawan A, Heryana N. *Prototype development of a low cost data logger and monitoring system for PV application*. 2016 3rd Conference on Power Engineering and Renewable Energy (ICPERE). Yogyakarta, Indonesia. 2016; 171-7.

[7] Cosmas CA, Dahlan NY. *Development of visual basic GUI for Option C energy saving of IPMVP*. 3rd IET International Conference on Clean Energy and Technology (CEAT) 2014. Kuching, Malaysia. 2014; 1-6.

[8] Rivai A, Rahim NA. *GUI for photovoltaic (PV) array monitoring system*. 3rd IET International Conference on Clean Energy and Technology (CEAT) 2014. 2014; 1-5.

[9] Dangare P, Mhizha T, Mashonjowa E. Design, fabrication and testing of a low cost Trunk Diameter Variation (TDV) measurement system based on an ATmega 328/P microcontroller. *Computers and Electronics in Agriculture*. 2018;148:197-206.