Gas Detector using Arduino and LabVIEW

Conference Paper *in* IEEE International Conference on Fuzzy Systems · November 2018

Bogdan Mihai
Lucian Blaga University of Sibiu

Gas Detector Using Arduino and LabVIEW

Mihai Bogdan

Computer Science and Electrical Engineering Department,
Lucian Blaga University of Sibiu, 550025, Romania,
E-Mail: mihai.bogdan[at]ulbsibiu.ro

Abstract

The objective of this article is to develop a technical project, to detect smoke and inflammable gases using an MQ-2 sensor, Arduino and LabVIEW graphical programming. Will be read the sensor analog output voltage and when the smoke reaches a certain level, it will make sound a buzzer and a red LED will turn on.

Hardware resources that will be used are: Arduino Uno, MQ-2 Smoke detection sensor, a breadboard, 5 mm red LED, Buzzer, 2 x suitable resistors for to limit the current through LED and Buzzer (220 Ohms is fine) and connecting wires. The results will be displayed through the serial interface on the computer, in the LabVIEW program.

Keywords: MQ gas sensors, Arduino, LabVIEW, LINX software, While loop, Case structure.

1 Introduction

The gas sensors of the MQ series are analog sensors, designed to detect the presence of different chemical components in the air. We can connect these devices with any microcontroller, like Arduino. There is a wide variety of MQ sensors. Each model is designed to detect one or more substances, designed for a specific use, such as flammable gas detection, air quality or detection of alcohol in breathed air.

The MQ-2 gas sensor, used in my project, are suitable for detecting LPG (Liquefied Petroleum Gas), propane, methane, alcohol, hydrogen, smoke. It is more sensitive to LPG and propane. Gas sensors are provided with both analog output and digital output. The analog output is increased the value of the voltage in proportion to the level of gas that is detected. The digital output can be used to detect gas leakage and hence trigger an alert system (for example a sound alarm or an sms activation). The digital output gives only two possible outputs – High and Low.

Due to its high sensitivity and fast response time of the MQ2 gas sensor, measurement can be taken as soon as possible. The sensitivity of the sensor can be adjusted by potentiometer.

The MQ gas sensors must be calibrated before making an accurate measurement. Even calibrated, these sensors do not have the guarantee to be part of a security system.

Despite its limitations, MQ gas sensors are widely used in various Arduino projects. For example, we can turn on or off a fan depending on air quality, we can make a small breath detector or an alarm to alert when smoke is detected.

The MQ sensors use small heater inside with an electro-chemical sensor that changes resistance in contact with the various substances. They are sensitive to a range of gasses and are used indoors at room temperature. The output is an analog signal

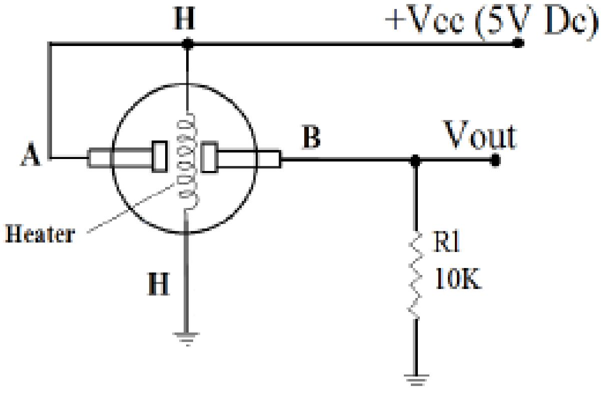


Figure 1.The schematic of the MQ gas sensor

and can be read with an analog input of the Arduino (https://playground.arduino.cc/Main/MQGasSensors).

Gas sensors are devices with high inertia, their response requires long periods of time to stabilize after a change in the concentration of the measured gases. This is due to the physical need of gas to leave sensitive material, which is a slow process.

All MQ models have a heater needed to increase the temperature of the sensor (Figure 1). As long as the heater does not reach the operating temperature, reading the sensor will not be safe. The heating time depends on each sensor model. In most models, a few minutes are enough, but some models require between 12-48 hours until stable measurements are obtained. On the other hand, each model needs its own voltage to power the heater. In many models this voltage is 5V.

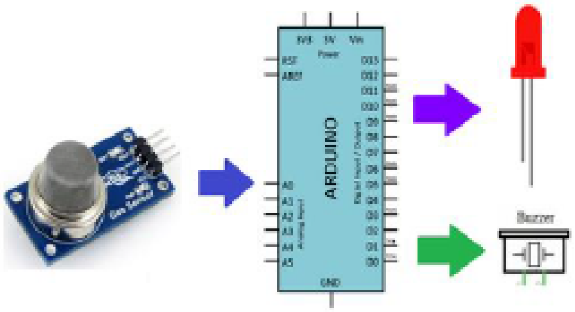


Figure 2.The components necessary for the elaboration of the technical project

2 The Virtual Instrument of the gas detector

I used the gas sensor by reading the analog values generated by it. I have connected the analog output of the sensor to the Arduino analog input A0. Then I used Arduino's digital outputs 5 and 8 to control the Buzer and the LED.

Represented in Figure 5 is the Front Panel of the VI. It contains the following controls and indicators:

- A control for setting the serial port to which Arduino is connected;
- A boolean Stop control to turn off the virtual instrument;
- A numeric Meter indicator to display the sensor output voltage;
- 2xLEDs indicators, that is turn on or off based on the voltage thresholds of the smoke;
- A control to set the analog pin to which the sensor is connected;
- Two controls to set the digital pins to wich the LED and Buzzer is connected

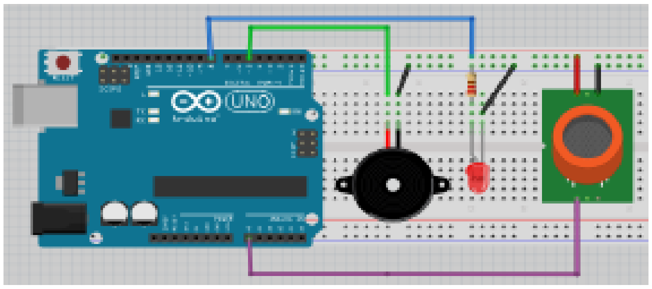
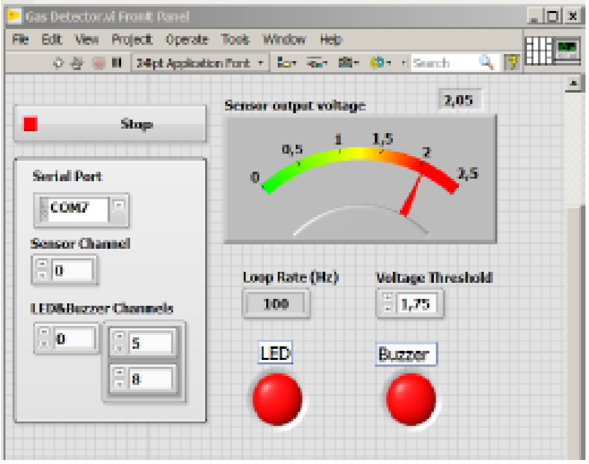


Figure 3. Circuit diagram of the Gas Detector


Gas Detector.vi Front Panel
File Edit View Project Operate Tools Window Help
24pt Application Font
Search
Stop
0
0,5
1
1,5
2
2,5
LED
Buzzer

Sensor output voltage 2,05

Serial Port COM7

Sensor Channel 0

LED&Buzzer Channels 0 5 8

Loop Rate (Hz) 100

Voltage Threshold 1,75

Figure 4. The Front Panel of the VI

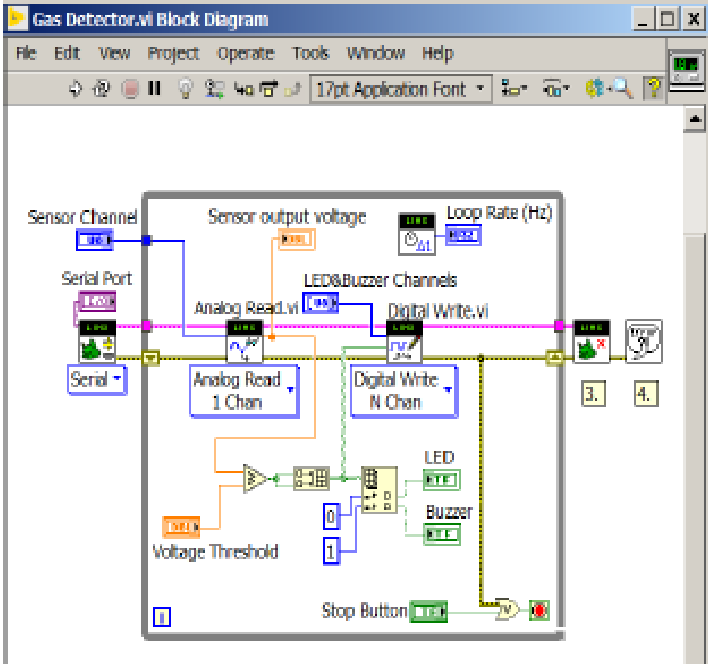


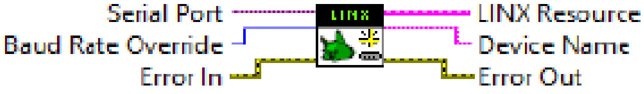
Figure 5. The Block Diagram of the VI

To make the Block Diagram of the VI (Figure 6), I used the LINX software, that is the the new way of programming Arduino. LINX provides easy to use LabVIEW VIs for interacting with common embedded platforms, like Arduino. The LINX functions operate in a similar manner like the LIFA function, but they have different names.

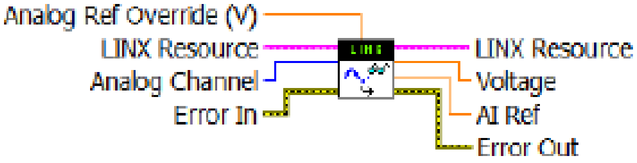
The Block Diagram of VI contains the controls and indicators terminals of the Front Panel, the various nodes, constants and the wires. The nodes in LabVIEW are different functions, subVIs and programming structures.

I used different LINX functions for programming Arduino platform:

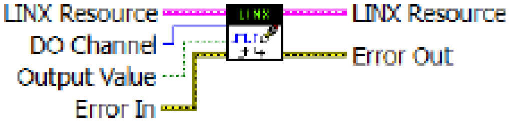
-The Open Function initiates communication. We can see the COM Port that Arduino is connected to. Each different program starts with the Open function;



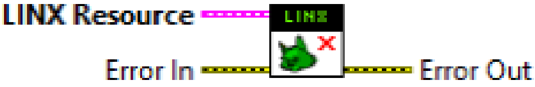
-The Analog Read Function read the value of the specified analog input channel, and return the sensor output voltage, which is compared with the voltage threshold set by the user;



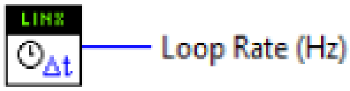
-The Digital Write Function allows us to select the digital input address that we will use on the Arduino. The Output Value input is active with the value being connected or sent;



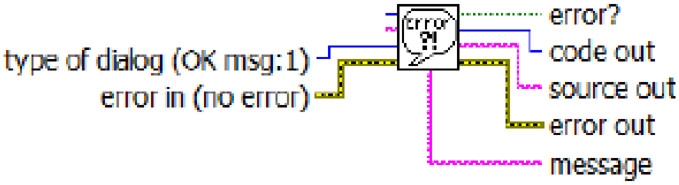
-The Close Function controls the end of the program. We must finish each different program with Close function;



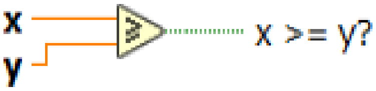
-With the Loop Frequency Function, we can see the loop rate in Hz.



-The Simple Error Handler Function indicates whether an error occurred. If an error occurred, this VI returns a description of the error and optionally displays a dialog box.



-The Greater Or Equal? Function returns TRUE if output sensor voltage is greater than or equal to voltage threshold. Otherwise, this function returns FALSE.



The Bock Diagram of the VI, also contains a While loop. A While Loop, executes the code it contains until a condition occurs (http://www.ni.com/getting-started/labview-basics/execution-structures).

Conclusion

The MQ2 gas sensor is used to detect gas leaks in small or large rooms and is a precautionary method for fires or intoxications.

The sensor has a high sensitivity and the main gases it targets are LPG, isobutane, propane, methane, alcohol, hydrogen and smoke. The sensor has a comparator so you can read analogue data in real time or find out if the gas concentration has exceeded a certain limit.

This small gas sensor detects the presence of fuel gas and smoke in concentrations of 300 to 10,000 ppm. It incorporates a simple analog voltage interface that only requires an analog input pin of the microcontroller.

Despite its limitations, MQ2 gas sensors are widely used in home electronics projects with Arduino. For example, we can turn a fan on or off depending on the quality of the air, make a small breathalyzer detector, or an alarm that sounds when smoke is detected.

References

Bogdan M., (2017): „Traffic Light Using Arduino Uno and LabVIEW", Proceedings of the 12th International Conference on Virtual Learning ICVL 2017, ISSN: 1844-8933, p. 286-290, October 28, 2017

https://www.arduino.cc

http://zone.ni.com/reference/en-XX/help/371361H-01/lvconcepts/intro_to_vis/

http://sine.ni.com/nips/cds/view/p/lang/ro/nid/212478

http://www.ni.com/getting-started/labview-basics/shift-registers

https://playground.arduino.cc/Main/MQGasSensors

http://www.ni.com/getting-started/labview-basics/execution-structures

https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MQ-7%20Ver1.3%20-%20Manual.pdf