Application End of Line Resistor (EOLR) and Voltage Divider for Passive Infrared Motion Sensor

Bibit Hartono

Department of Mechanical Engineering, Swiss German University, Tangerang 15143, Indonesia

Abstract. End of Line Resistor (EOLR) is a resistor that has a certain value to protect the circuit at the end of a loop or zone. The End of Line Resistors is used in security systems to enable the control panel to monitor wiring of open and closed. In this research using an Arduino microcontroller and Passive Infrared (PIR) Motion Sensor that connected with the systematic circuit to supervisory a loop or zone. Systematic circuit use combination of the resistor as the end of line resistor and voltage divider method. Analog input on Arduino uses to detect different states of sensors. The 1K resistor in series with the sensor line provides some protection for the input pin against the unexpected current flow. The 4K7 pull-up resistor, when combined with the resistance provided by the sensor, acts as a voltage divider, exposing the analog input to a different voltage depending on the state of the sensor. The result of measurement viewed by serial monitor function on the personal computer. PIR Sensor connected with an analog pin on microcontroller running well and there is no significant voltage value changing.

Keyword: Microcontroller, Motion Sensor, Resistor, Voltage Divider, Analog

1. Introduction

Resistors are basic electronic components used to limit the amount of current flowing in a circuit. Another function of the resistor is as a limiting electric current can be used as a divider of electrical voltage and as a decrease in the voltage of the electric current. In the digital circuit, the "high" signal is 5 volts and the "low" signal is 0 volts, whereas in the digital circuit 3.3 volts the "high" signal is 3.3 volts and the "low" signal is 0 volts. Of course, the "high" signal does not have to be exactly 5 volts or 3.3 volts, depending on the tolerance of the circuit and Integrated Circuit used. In a digital circuit, when using the sensor as the input data to the microcontroller sometimes occurs the problem of unreadable value. The input value floats between high and low. To solve this problem you can use a pull-up or pull-down resistors that use the voltage divider principle.

In this paper, it is observed that a systematic design approach for interfacing the Passive Infrared Motion Sensor to a single port of a microcontroller is possible to improve the security system.

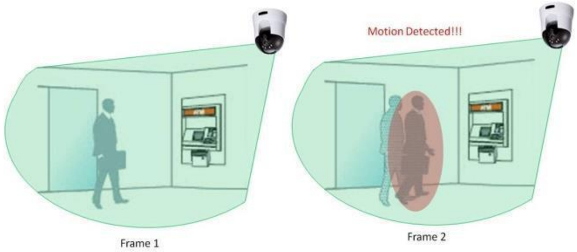

Figure 1. Motion Sensors

2. Scope

The scope of this study is limited at the Passive Infrared (PIR) motion sensor and Arduino microcontroller for the security home security system. In this research use Arduino's analog inputs to build sensor circuits that are as fully featured and advanced. The System allows detecting any of possible failure, tamper, or trigger conditions automatically with 4 channel sensors.

3. Methodology

Research use hardware and software design method the following below:

3.1. Hardware Design

Research use a notebook and components according to the following table:

Table 1. Component required

No	Components	Qty	No	Components	Qty
1	Arduino	1	6	Diode 1N4001	4
2	PIR Sensor	4	7	Relay 12Volt	8
3	Resistor 4K7Ω	17	8	Buzzer	1
4	Resistor 1KΩ	4	9	LED	8
5	Resistor 680Ω	8	10	Power Supply 12Volt	1

3.2. Software Design

Source code for main program to show the result via serial monitor function can be seen below:

```
// Show the result via serial monitor function
Serial. print(sensorInput, DEC );
Serial. print( ": " );
Serial. print(voltageReading, 2);
Serial. print( " (" );
Serial. print(valueReading, DEC);
Serial. print( " (" );
Serial. print(state, DEC );
Serial. print( ") | " );
```

3.3. Design Justification

Design according to the schematic circuit as shown by following schematic diagram:

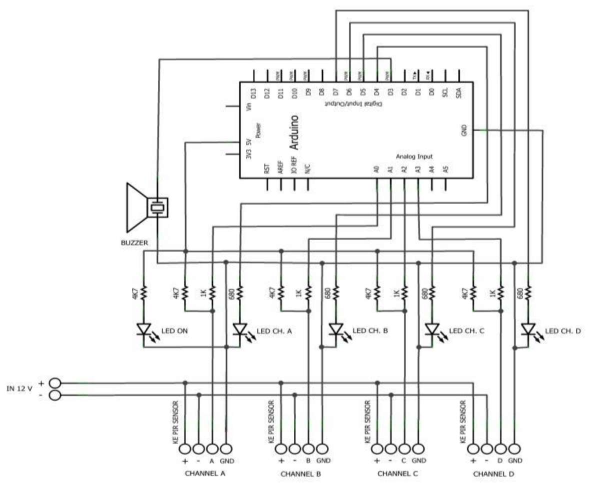

Figure 2. Control panel schematic diagram

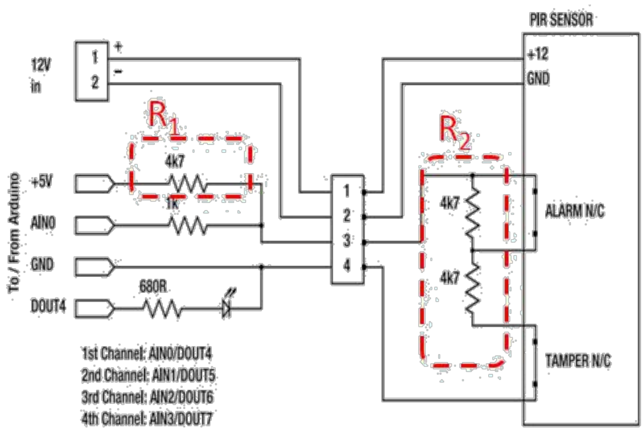

Figure 3. Sensor schematic diagram

3.4. Design of Experiment
Wiring all of component is shown by the following figure:

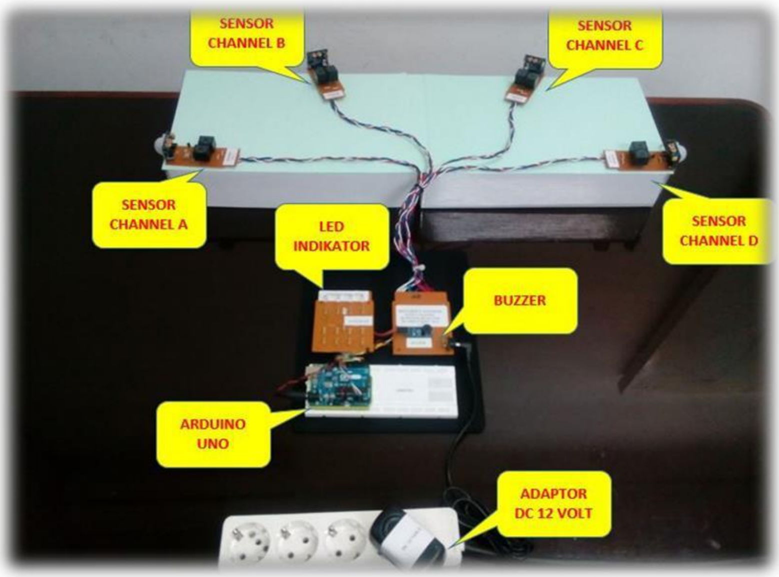

Figure 4. Wiring component

4. Test Result

4.1. Test Result on State 0 - wire shorted

According to the circuit on figure 3, the calculations using voltage divider formula:

$$= \frac{\quad 2}{(\ 1 + \ 2)} 5$$

$$= \frac{\sim 1}{(4\ 7 + \qquad \sim 1)} 5$$

$$= \sim 1 \qquad h \quad h\, 1 : 0.00488 = \sim 200$$

Test result on state 0 or when wire shorted is shown by the following figure below:

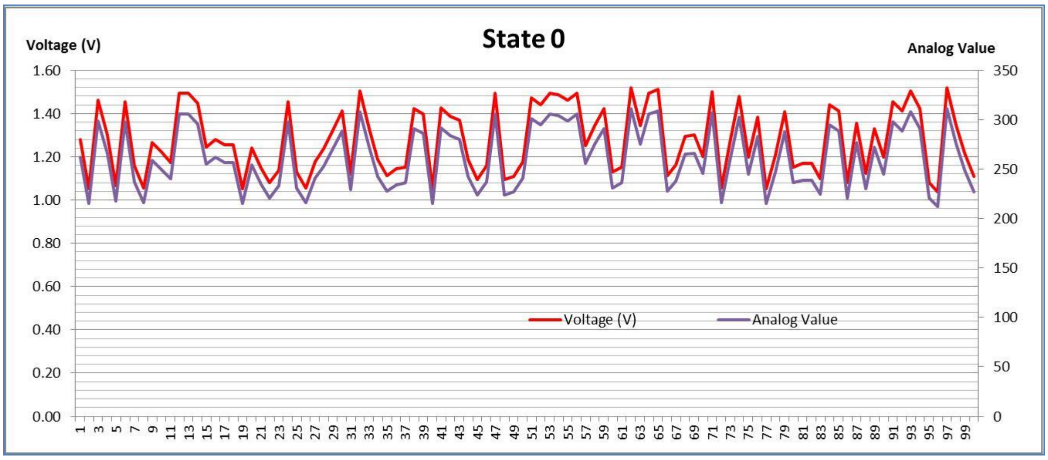

Figure 5. Test result on state 0

According to figure 5, from measure until 100 times, it can be seen that the maximum voltage is 1.52 volt with analog value is 310 and the minimum value is 1.04 volt with analog value is 213. There is no significant voltage value changing in the experiment.

4.2. Test Result on State 1 – normal condition or no trigger

According to the circuit on figure 3, the calculations using voltage divider formula:

$$= \frac{_2}{(_1 + _2)} 5$$

$$= \frac{4\ 7}{(4\ 7 + \quad 4\ 7)} 5$$

$$= 2.5 \qquad h \quad h\ 2.5 : 0.00488 = 512$$

Test result on state 1 or when a normal condition is shown by the following figure below:

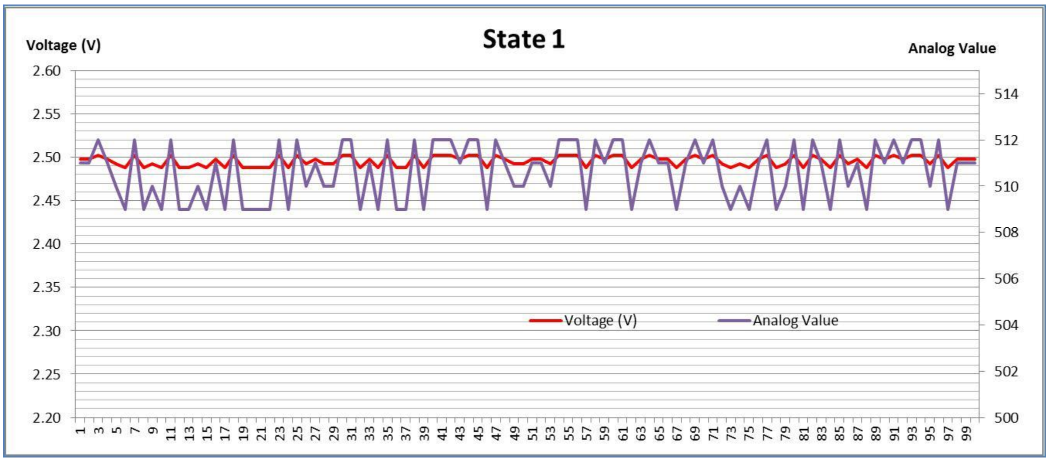

Figure 6. Test result on state 1

According to figure 6, from measure until 100 times, it can be seen that the maximum voltage is 2.50 volt with analog value is 512 and the minimum value is 2.49 volt with analog value is 509. There is no significant voltage value changing in the experiment.

4.3. Test Result on State 2

According to the circuit on figure 3, the calculations using voltage divider formula:

$$= \frac{{}_{2}}{({}_{1} + {}_{2})} \quad 5$$

$$= \frac{4\ 7 + 4\ 7}{(4\ 7 + (4\ 7 + 4\ 7))} \quad 5$$

$$= 3.33 \qquad h \quad h\, 3.33 : 0.00488 = 680$$

Test result on state 2 or when the sensor tripped is shown by the following figure below:

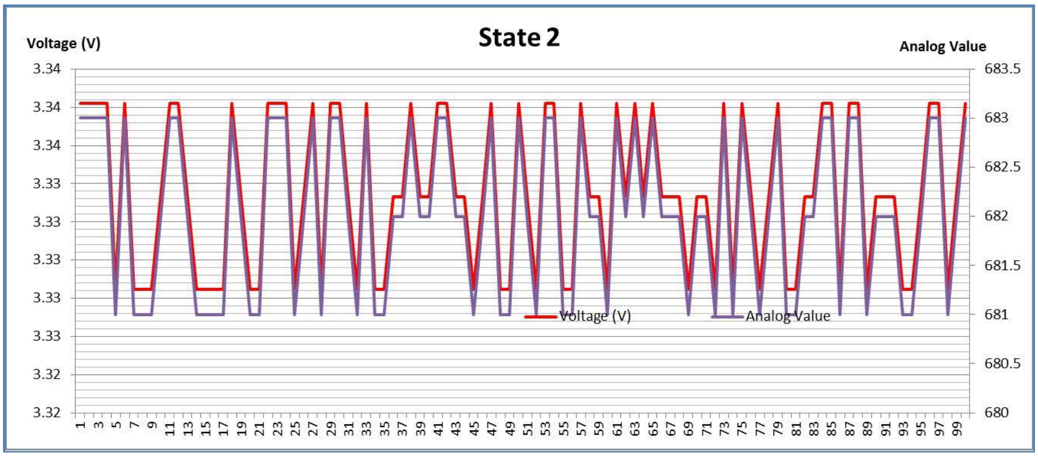

Figure 7. Test result on state 2

According to figure 7, from measure until 100 times, it can be seen that the maximum voltage is 3.34 volt with analog value is 683 and the minimum value is 3.33 volt with analog value is 681. There is no significant voltage value changing in the experiment.

4.4. Test Result on State 3

According to the circuit on figure 3, the calculations using voltage divider formula:

$$= \frac{{}_{2}}{({}_{1} + {}_{2})} \quad 5$$

$$= \frac{}{(4\ 7 + \quad)} \quad 5$$

$$= 5 \qquad h \quad h\, 5 : 0.00488 = 1023$$

Test result on state 3 or when wire cut or tamper detected is shown by the following table below:

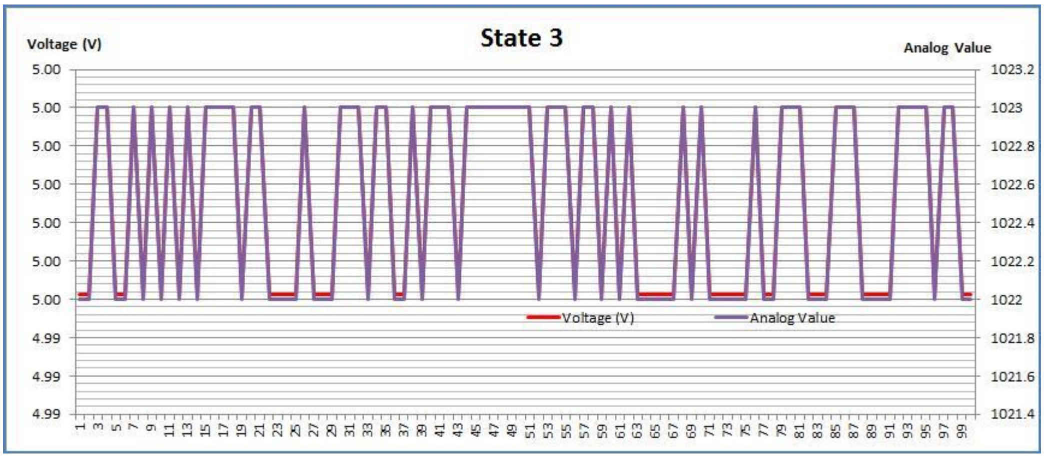

Figure 8. Test result on state 3

According to figure 8, from measure until 100 times, it can be seen that the maximum voltage is 5 volt with analog value is 1023 and the minimum value is 5 volt with analog value is 1022. There is no significant voltage value changing in the experiment.

From above test results can be summarized states or sensor conditions a voltage value and analog value according to the following table 2:

Status - Condition	***Resistance (Ohm)***	***Voltage (Volt)***			***Analog Value***		
		Calculation	*Actual Minimum*	*Actual Maximum*	*Calculation*	*Actual Minimum*	*Actual Maximum*
0	*0*	*~1*	*1.04*	*1.52*	*0*	*213*	*310*
1	*4K7*	*2.5*	*2.49*	*2.5*	*512*	*509*	*512*
2	*9K4*	*3.3*	*3.33*	*3.34*	*675*	*681*	*683*
3	*Infinite*	*5*	*5*	*5*	*1023*	*1022*	*1023*

Table 2. State, voltage and analog value of system

5. Conclusion

1. Analog input on Arduino uses to detect different states of sensors and there is no significant voltage value changing in the experiment.
2. The End of Line Resistor works to supervise the field wiring for open or short circuit conditions and report to the control panel.
3. This design gives four possible voltage levels applied to the Arduino analog input depending on the state of the sensor.

4. The system can detect 4 states of the sensor as the following:
    - the wire has been short-circuited then resistance value is 0□ and the voltage is ~1 Volt
    - normal conditions or no trigger then resistance value 4K7□ and the voltage 2.5 Volt.
    - the sensor has been trigger then resistance value 9K4□ and the voltage value is 3.3 Volt.
    - the wire of the system has been cut then resistance became infinite and voltage is 5 Volt.
5. The difference between calculation and the result caused by ideally the switch has zero series resistance and the microcontroller input pin has an infinite input impedance. But in fact, the switch circuit has a non-zero series resistance and the microcontroller input pin has an infinite input impedance.

References

[1] John Boxall. Arduino Workshop - A Hands-On Introduction with 65 Projects. ISBN-10: 1-59327-448-3. United States of America
[2] Jonathan Oxer and Hugh Blemings. Practical Arduino - Cool Projects for Open Source Hardware. ISBN-13:978-1-4302-2477-8. 2009. United States of America
[3] Hanumantea, Sandeep, and Sawlanib, Kirti. (2011). Effective Utilization of a Microcontroller Port for Optimisation of Hardware. Faculty Department of Electronics Engineering, K. J. Somaiya College of Engineering. India
[4] Swapnika K. Shinde and Amol B. Jagadale. (2016). Enhancing Optimization and Device Portability by Minimizing Port Pin Count. International Conference on Communication and Signal Processing. India.
[5] Winkler, Fabian. (2007). Arduino Workshop. Spring
[6] Microchip Technology. (2009). Compiled Tips N Tricks Guide. Microchip Technology Inc. Pp.1-137
[7] J. Julichr. (2003). Hardware Techniques for PIC micro Microcontrollers. Microchip Technology Inc. Pp. 1-12
[8] Honeywell. (2010). Fire Detection System Guides. United States