

Design and Implementation of a Non-Contact Detection Sensor with Parasitic Capacitance and Pull-up Resistance

Joon-Hyuk Choi, Hak-Hui Choi

Abstract— This paper designs and implements a capacitive non-contact detection sensor using parasitic capacitance and pull-up resistance. The proposed sensor replaces the peripheral components of existing sensors with the parasitic capacitance and pull-up resistance inherent in the Arduino, thereby reducing the sensor design cost. It also provides stable output by using experimentally designed compensation coefficients and filters. Additionally, the measured sensor values are monitored via a mobile application, and the performance of the implemented sensor is evaluated by comparing the measured values with the actual capacitor values.

Keywords—MCU, Arduino, App Inventor, Parasitic Capacitance, Pull-up Resistance

I. Introduction

A sensor is a device that detects and measures the amount, movement, temperature, sound, light, electromagnetic waves, etc., of an object, as shown in Figure 1. With the development of the sensor industry, ongoing research is being conducted to reduce manufacturing costs and miniaturize sensors.

Among them, a capacitive non-contact detection sensor measures the change in capacitance between conductors(metal wires) with a fixed distance in real time to detect the presence or movement of objects near the sensor. Like other sensors, existing sensors achieve accurate and stable measurements by adding peripheral devices or using mechanical compensation. However, these methods increase additional costs and the size of the sensor system, reducing the usability of the sensor.

To overcome this problem, this paper implements a capacitive non-contact detection sensor without using additional peripheral devices, by utilizing the parasitic capacitance and pull-up resistance inherent in the Arduino's MCU (Atmega328P)^[1].

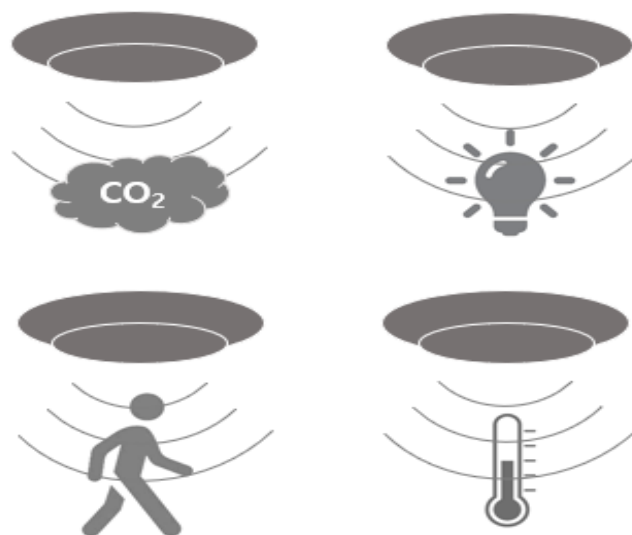


Figure 1. Types of Sensors

manufacturing costs, and minimizes the sensor's size, enhancing its applicability across various applications.

After implementing the sensor, the capacitor's capacitance is measured and compared with the values written on the capacitor to evaluate the sensor's performance. A mobile application based on APP Inventor is developed to monitor the real-time measured capacitance values.

Joon-Hyuk Choi
Mosan Elementary School
Republic of Korea
Hak-Hui Choi
Hyundai Motor Company
Republic of Korea

This approach minimizes the number of peripheral elements required for human or object sensing. it also reduces

II. Main Content

This paper designs and implements the proposed capacitive non-contact detection sensor using the Arduino UNO R3.

A. Proposed Sensor Design

The proposed sensor has a structural difference compared to existing sensors, as shown in Figure 2^[2].

The capacitance value to be measured (C_{Target}) is calculated using the parasitic capacitance ($C_{Parasitic}$) between A1 port and GND as follows in Equation 1.

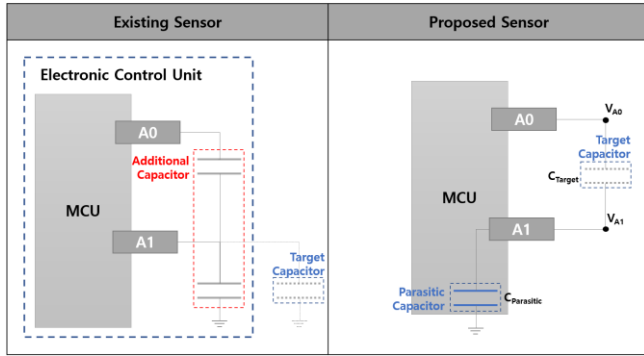


Figure 2. Structural Differences of between Existing and Proposed Sensor

$$V_{A1} * (C_{Target} + C_{Parasitic}) = V_{A0} * C_{Target}$$

...(Equation 1)

Before calculating C_{Target} , to obtain $C_{Parasitic}$, V_{A0} outputs an analog value of 5V (1023) from port A0, replacing C_{Target} with a specific value (100nF), and the measured Analog Value of V_{A1} (822) at port A1 is substituted into the equation, resulting in Equation 2.

$$C_{Parasitic} = \frac{V_{A0} * C_{Target}}{V_{A1}} - C_{Target}$$

$$\doteq 24.45 pF$$

...(Equation 2)

Substituting this into Equation 2 and replacing the voltage analog values with those measured from the analog port results in Equation 3.

$$C_{Target} = \frac{V_{A1} * C_{Parasitic}}{V_{A0} - V_{A1}}$$

$$= \frac{Analog Value(V_{A1}) * 24.45 pF}{1023 - Analog Value(V_{A1})}$$

...(Equation 3)

However, the method of calculating C_{Target} from Equation 3 will result in significant errors when $C_{Parasitic}$ is much smaller than C_{Target} .

If C_{Target} is about 1nF (analog input value of $V_{A1} \approx 977$) or more, the charging time (t_{Charge}) based on the pull-up resistance ($R_{Pull-up}$) at the A0 port and the C_{Target} value is used to measure C_{Target} , as shown in Figure 3. Equation 4 represents the relationship between $R_{Pull-up}$, C_{Target} , and t_{Charge} over time.

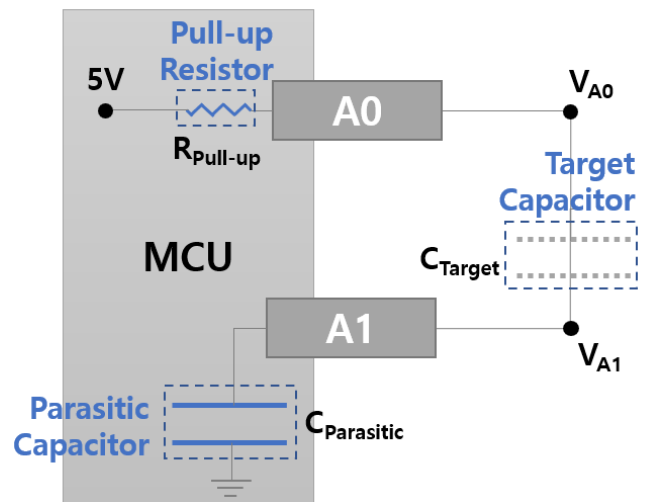


Figure 3. Capacitance Measurement Circuit Based on $R_{Pull-up}$

...(Equation 4)

$$V_{A0}(t_{Charge}) = V_{Pull-up} (1 - e^{-\frac{t_{Charge}}{R_{Pull-up} * C_{Target}}})$$

In Equation 4, $V_{Pull-up}$ is set to 5V (1023) by configuring the A0 pin to INPUT_PULLUP, and the value of $R_{Pull-up}$ is measured between the A0 pin and GND using a multimeter ($I_{Multimeter}$), then substituted into Equation 5.

$$R_{Pull-up} = \frac{V_{Pull-up}}{I_{Multimeter}}$$

...(Equation 5)

Substituting $V_{Pull-up} = 5V$ and $I_{Multimeter} = 0.143mA$ into this equation gives $R_{Pull-up} = 34.8k\Omega$. This $R_{Pull-up}$ value and the fixed $V_{Pull-up}$ are substituted into Equation 4, which is then rearranged to solve for C_{Target} , replacing the voltage value with the measured value from the analog port, resulting in Equation 6.

$$C_{Target} = \frac{t_{Charge}}{34.8k * \ln(1 - \frac{Analog Value(V_{A0}(t_{Charge}))}{1023})}$$

...(Equation 6)

By substituting the 63% of 5V (1023), which is 3.15V (645), and measuring the time it takes to reach this value, the C_{Target} value can be calculated.

However, the C_{Target} value obtained through this process may be unstable due to the minimal number of components

used in the sensor. To address this, the proposed sensor outputs a stabilized result by averaging the three middle values of five samples, excluding the minimum and maximum values, and applying a compensation coefficient.

The method for calculating C_{Target} using $C_{Parasitic}$ and $R_{Pull-up}$ in this section can be summarized in the flowchart shown in Figure 4.

B. Proposed Sensor Implementation

The proposed sensor is implemented as shown in Figure 5.

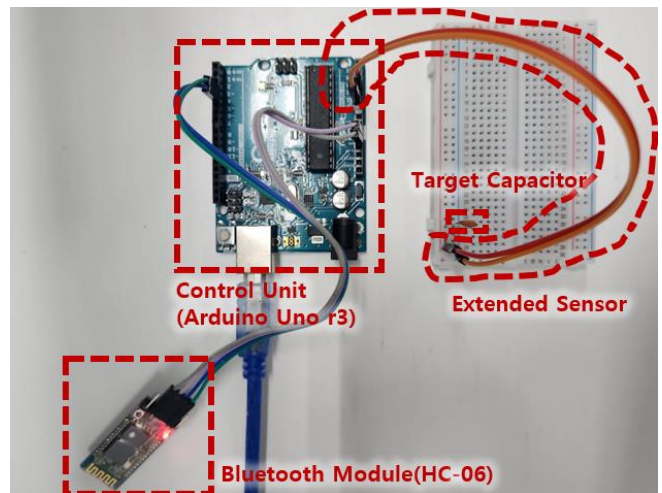
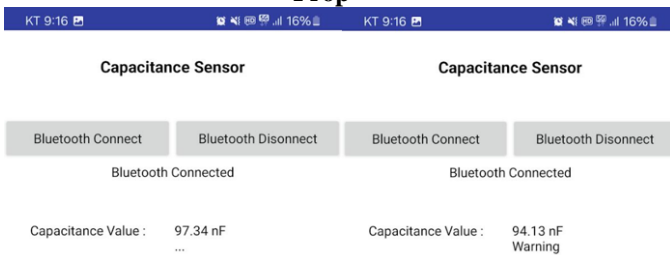
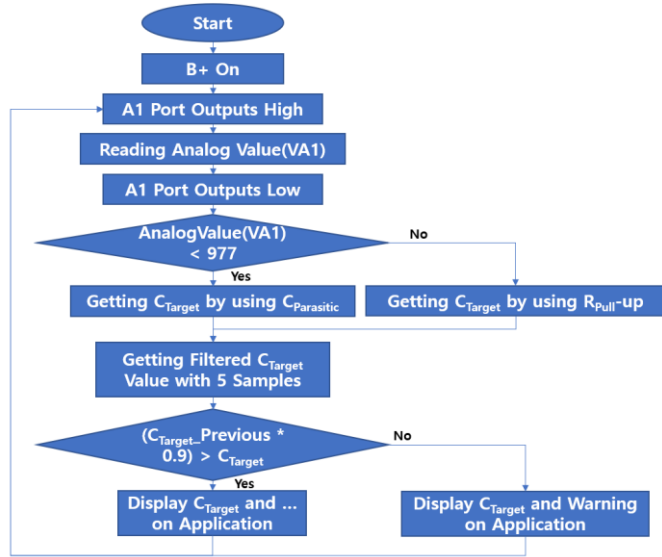


Figure 6. Mobile Application for Monitoring C_{Target}

Figure 4. Flowchart of How to Obtain C_{Target} on Prop



The proposed sensor consists of an Arduino Uno R3 for control, a sensor extension (wire) that can detect human or body proximity, and a Bluetooth module (HC-06) to transmit the measured data to a mobile device



e. The sensor extension is used in a form where the insulation is in contact with each other to minimize the change in capacitance. The shape and length of the extension can be adjusted depending on the application in which the sensor is used. Since the sensor extension also has its own inherent capacitance, a compensating parameter ($C_{ExtendedLine}$) is applied, as shown in Equation 7.

$$C_{Target} = \frac{t_{Charge}}{34.8k * \ln(1 - \frac{Analog Value(V_{A0}(t_{Charge}))}{1023})} - C_{ExtendLine}$$

...(Equation 7)

The measured sensor data is transmitted to the mobile application via the Bluetooth module in 4-byte (float) format.

The mobile application applies an averaging filter and compensation coefficients to the data received from the



Figure 7. Measured Capacitor Samples

Arduino, then displays it on the screen as shown in Figure 6.

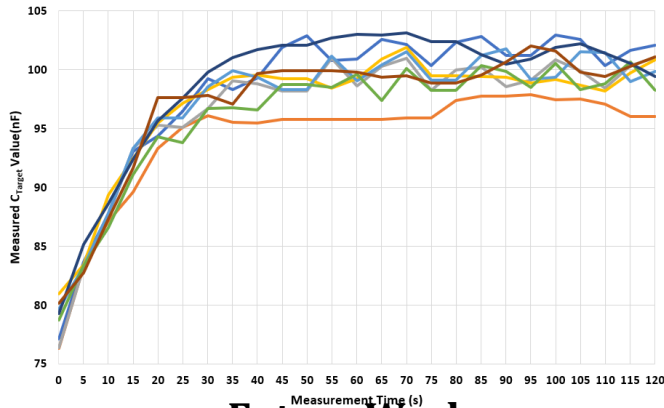
The left image of Figure 6 shows the idle state, and the right image shows the presence of a body or object near the sensor, displaying a "Warning"^[3].

III. Conclusion

The sensor designed and implemented in Chapter 2 is verified using eight commercially available ceramic capacitor components (100nF), as shown in Figure 7.

The capacitance values measured using the 8 samples are shown in Figure 8. The measured capacitance values gradually converge to the value specified on the capacitor component (10nF) after an initial stabilization period.

Figure 8. Measured



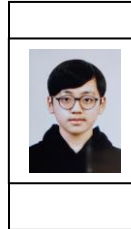
iv. Future Works

The non-contact detection sensor implemented in this paper aims to overcome the stabilization time issue identified during verification. In the next phase, the sensor will be designed not only to detect the presence of a body or object but also to output accurate capacitance values without stabilization time. To achieve this, the compensation coefficient used during the design phase will be changed from a constant to a variable based on time, and more capacitor samples will be used to ensure reliability in the verification stage.

References

- [1] F. Reverter, M. Gasulla and R. Pallas-Areny, "A low-cost microcontroller interface for low-value capacitive sensors," Proceedings of the 21st IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No.04CH37510), Como, Italy, 2004, pp. 1771-1775 Vol.3, doi: 10.1109/IMTC.2004.1351425
- [2] F. Reverter and O. Casas, "Direct Interface Circuit for Differential Capacitive Sensors," 2008 IEEE Instrumentation and Measurement Technology Conference, Victoria, BC, Canada, 2008, pp. 1609-1612, doi: 10.1109/IMTC.2008.4547300.
- [3] T. Munasinghe, E. W. Patton and O. Seneviratne, "IoT Application Development Using MIT App Inventor to Collect and Analyze Sensor Data," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 6157-6159, doi: 10.1109/BigData47090.2019.9006203.

About Author (s):



Joon-Hyuk Choi
 Mosan Elementary School, Republic of Korea
 Interested in [Design and Implementation Electronic System]



Hak-Hui Choi
 Hyundai Motor Company, Republic of Korea
 Interested in [Design and Implementation Automotive ECU]