PIR Motion Sensor Alarm using Arduino MKR Wi-Fi 1010

Pranav Kumar Jha[1], Krupa Sunilbhai Vora[2] and Fatemeh Eslaminasab[3]
[1,2,3]*Department of Electrical and Computer Engineering*
Concordia University, Montreal, Canada
[1] Student ID: 40081750, jha_k.pranav@live.com
[2] Student ID: 40171212, krupavora1998@gmail.com,
[3] Student ID: 40170674, Fatemeheslaminasab1996@gmail.com

***Abstract*—The demand for smart home security systems is increasing nowadays since crime is rampant, so they are utilized to warn owners about unusual conditions that could occur while they are away from their homes. A wireless motion detection sensor system using the Arduino platform is proposed. In this system, sensory data is gathered from the surrounding environment by a PIR motion sensor, and Arduino interprets it for proper action. In case of undesired activities and unwanted conditions, the alarms are activated; in addition, homeowners will be notified through the user's application. Arduino MKR Wi-Fi 1010 is used in this project, and it has a built-in Wi-Fi module that allows a connection between the user's application and the motion detection sensor system.**

***Index Terms*—Arduino, PIR, Home Security, IoT Cloud, Webhooks, IFTTT**

I. Introduction

Nowadays, smart home systems, which are based on sensors, are in increased demand. Smart home systems, especially home security systems, have been widely utilized since there are many criminal offenses such as invasion, theft and house breakage. Consequently, homeowners search for approaches that provide the safety of their homes. These systems monitor the home, detect undesired activities and conditions, and notify homeowners when they are far away. Smart home systems aim to gather and interpret sensory data from the surrounding environment and give homeowners live updates.

The application of a wireless motion detection sensor system using the Arduino platform is proposed in this project. The home warning system consists of a microcontroller with some compatible sensors, actuators, etc. Most home security systems utilize motion sensors along with other sensors. The PIR sensor is used in this project, the most widely used motion sensor in such systems. Arduino processes the data, which is gathered by sensors, and activates an alarm when the PIR sensor detects movement and the alarm will be activated. But how would the residents get the notification if they are not home? Since the alarm system does not have a connection with the users. There will be a user application. The application and the alarm system are connected via Wi-Fi module in Arduino. So it allows users to receive notifications through the application in their smartphone via Wi-Fi.

The rest of the report is structured as follows. Section II illustrates the most relevant works for this project. Section III demonstrates the methodology, hardware specification, and design of the proposed system. The results from software and hardware implementation are presented in Section IV. Finally, the conclusion of the project is given in Section V.

II. Related Works

Several methods have been proposed for home security using Arduino Uno [1], [2], and Arduino Mega [3]. They have to use a separate Wi-Fi module to send notifications to users which increases the costs of the application. Application of a more secure Arduino MKR Wi-Fi 1010 board with built-in Wi-Fi, which has built-in Wi-Fi, in smart home systems is yet to be discovered. We are creating a motion sensor alarm using a PIR sensor, an Arduino MKR Wi-Fi 1010 board, IFTTT (If This Then That) integration to webhooks to trigger an email notification or call to the users in real time. The user receives an email with the relevant information or a phone call to the cell phone of the user whenever a motion is detected by the PIR sensor in the proximity of 20 cm to 5 meters.

III. Methodology

A. Components used for the project

The hardware parts that are used in this project are [4]

- *Arduino MKR Wi-Fi 1010 Board*: The Arduino board as shown in Fig. 1 [4] uses an Arm® Cortex®-M0 32-bit SAMD21 processor. The SAMD21 from Atmel is a low-power, high-performance flash microcontroller based on the ARM® Cortex®-M0+ microchip. This offers 256KB Flash and 32KB SRAM, 48MHz Operating Frequency6 communication modules (SERCOM) that can be configured as universal asynchronous receiver-transmitter (UART)/universal synchronous/asynchronous receiver/transmitter (USART), Serial Peripheral Interface (SPI), 20 x Pulse Width Modulation (PWM) channels. PWM, is used to get analog results into digital form done with digital control by creating a square wave, a signal switched between on and off [4]. This also offers a full-speed USB device and an embedded host 1.62V to 3.63V power supply. This microcontroller is very powerful compared to the older 8-bit/16MHz microcontrollers from Arduino [5], [6]. The comparison of different Arduino

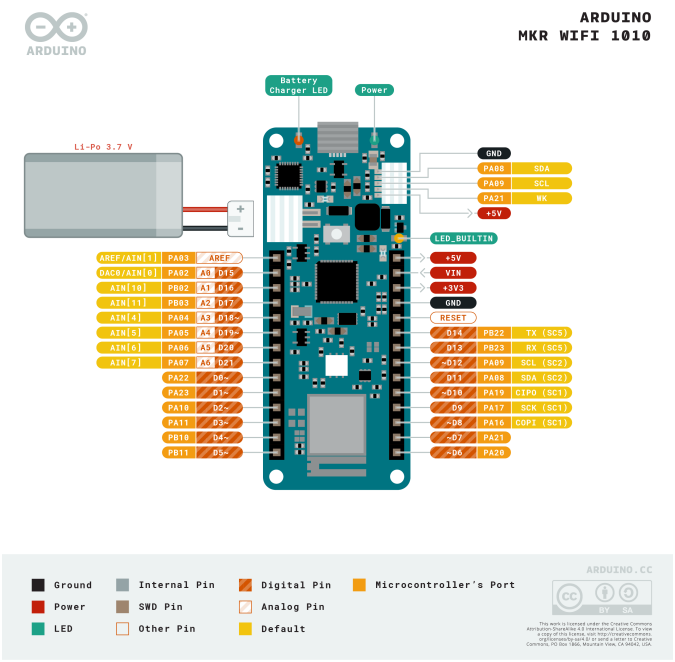


Fig. 1: Arduino MKR Wi-Fi 1010 Board

boards with the Arduino MKR Wi-Fi 1010 is provided in Table I.

TABLE I

	Arduino Uno R3	Arduino Mega 2560	Arduino MKR 1010 Wi-Fi
Microcontroller	ATmega328P	ATmega2560	SAMD21 Cortex-M0+32 bit low-power ARM
Clock Speed	16 MHz	16 MHz	48 MHz
Flash Memory	32 kB	256 kB	256 kB
EEPROM	1 kB	4 kB	None
SRAM	2 kB	8 kB	32 kB
Digital I/O Pins	14	54	22
PWM Digital I/O Pins	6	15	12
Analog Input Pins	6	16	8
Shield Connectivity	Yes	Yes	Speciality Shields
Wi-Fi	No	No	Yes

- Passive InfraRed (PIR) Sensor: This will be used to detect motion around the house or the premises where the alarm system is installed.
- 330 Ω Resistor
- Light-Emitting Diode (LED): A red LED is chosen for our project.
- Universal Serial Bus (USB) Cable: This connects Arduino board to our computer.
- Connecting Wires: These are used to make connections on the Breadboard.

B. Arduino IoT Cloud

The Arduino IoT Cloud allows to create IoT projects where we can make configuration, write and upload code to the board. Device Configuration is done here which will be used to collect data and control the equipment. A cloud variable is synced between Arduino board and the Arduino IoT Cloud. Creating and configuring of the variables are done by creating the 'Things' inside the IoT Cloud and set the variable info there, the Thing in our case is motion which is a boolean function. The variable permission is set to read-only as shown in Fig. 2. 'Arduino create agent' is used to upload code to the board using a web editor in the IoT cloud. Fig. 3 shows the associated device online inside the Arduino IoT Cloud when the experiment was being done.

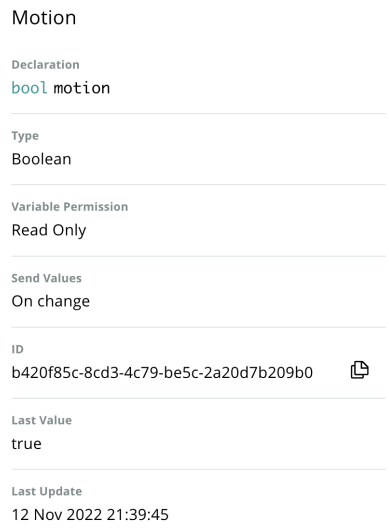


Fig. 2: Things created as 'Motion' in the Arduino IoT Cloud

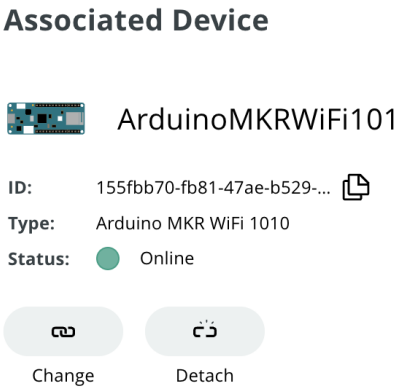


Fig. 3: Status of Arduino MKR Wi-Fi 1010 board in the Arduino IoT cloud while the alarm system is operational and running

The physical connections to the Arduino board and other components on the breadboard are shown in Fig. 5. The circuit connection to the Arduino pins and PIR pins are shown in Table II and the connections to the Arduino pins and LED pins are shown in Table III.

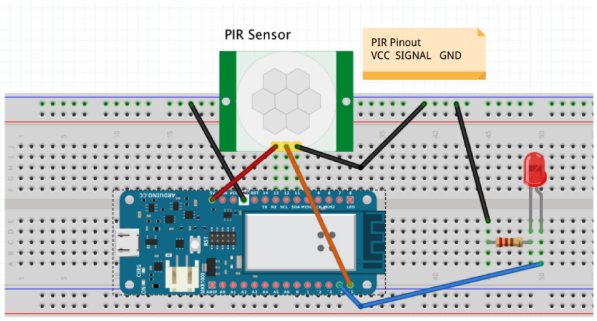


Fig. 4: Set up of the circuit on the breadboard

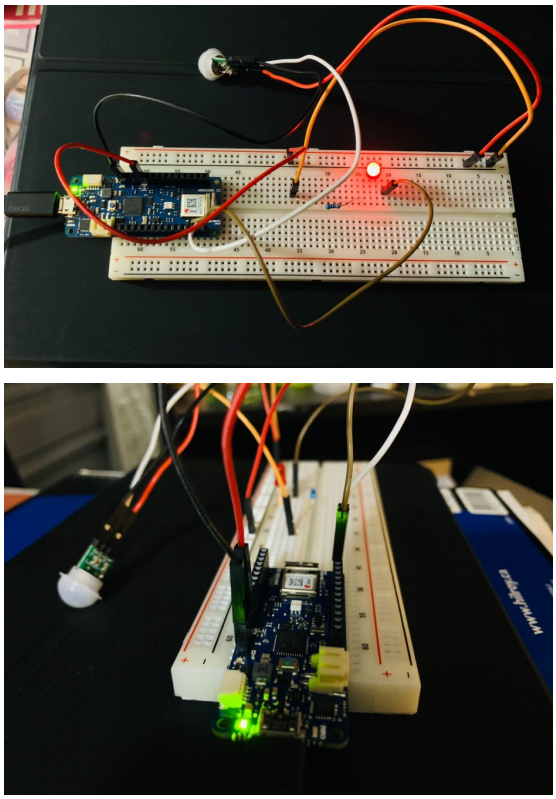


Fig. 5: Circuit connections shown on the breadboard

C. Webhooks integration with IFTTT

1) Webhooks: This is an event-driven application feature built into the Arduino IoT clouds enabling us to send and receive automated services to and from other services [4]. The webhook integrated with our Arduino IoT Cloud is active after setting it up as shown in Fig. 6.

Fig. 6: Active webhooks on Arduino IoT

2) IFTTT: IFTTT stands for "If This Then That." It helps users to automate web-based tasks and can connect to various devices, services or apps by creating "applets" to perform automated services such as email, call, text, etc [7]. The

TABLE II: PIR Connections to the Arduino MKR Board

Arduino MKR Pin	PIR Pin
VCC	5V
GND	GND
5	Signal

TABLE III: LED Connections to the Arduino MKR Board

Arduino MKR Pin	LED Pin	Connections
4	LED Anode	
	Signal	Resistor
5	Signal	

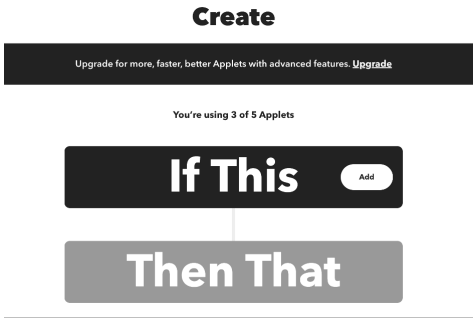


Fig. 7: IFTTT Services

applets for sending emails and calls are shown below in Fig. 8 and Fig. 9, respectively.

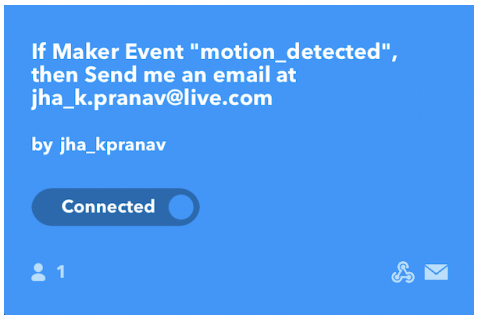


Fig. 8: Applet for Email

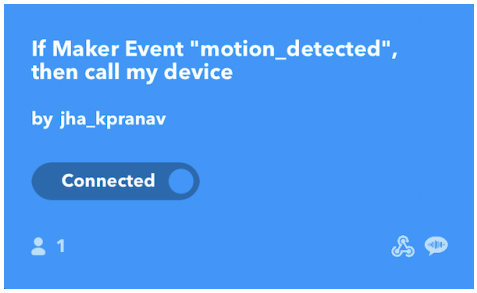


Fig. 9: Applet for Call

IV. Results

A. Code and Output

Fig. 10 shows the code to perform the action for the alarm system. When a motion is detected by the PIR sensor, it sets

the *val* to *HIGH* which was initially set to zero and this turns the LED *ON*. Now, as the property of the variable *motion* is a boolean, we check if the *motion* is initially set to *LOW* or not. if it is *LOW*, we set it to *HIGH*. Next, the "*Motion is being Detected*" is printed on the monitor screen of the Arduino IoT cloud. And, if the *motion* is *LOW*, we set it to *HIGH*. Further, if the motion stops, it turns the LED *OFF*, prints the output as "*Motion has Ended*", and again, if the *motion* is *HIGH*, we set it to *LOW*. The Full source code is provided in Appendix A.

```
  pinMode(ledPin, OUTPUT);
  pinMode(inputPin, INPUT);

}

void loop() {
  ArduinoCloud.update();
  //Your code here

  val = digitalRead(inputPin); // reading input value
  if (val == HIGH) {          // checking if the input is high or not
    digitalWrite(ledPin, HIGH); // if high then turning the led ON
    if (motion == LOW) {
      Serial.println("Motion is Detected in your home!"); //printing the change in output
      motion = HIGH;
    }
  }
  else {
    digitalWrite(ledPin, LOW); \\ turning the LED OFF
    if (motion == HIGH) {
      Serial.println("Motion has Ended!"); //printing the chnage in output
      motion = LOW;
    }
  }

}
```

Fig. 10: Code for the Alarm system on the Arduino IoT Cloud

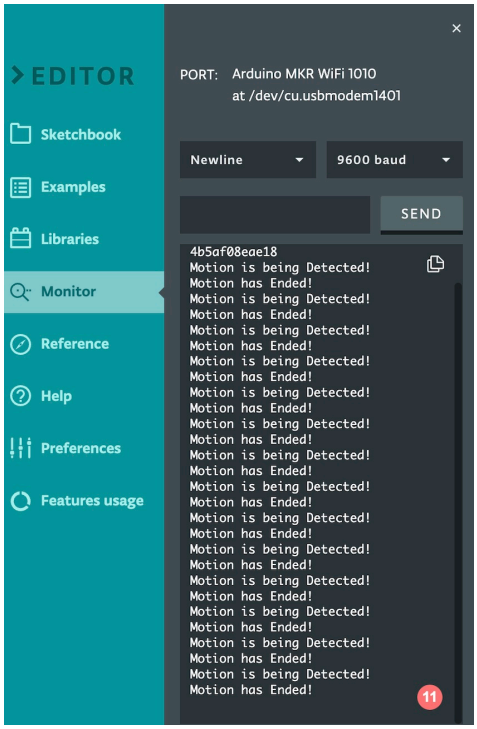
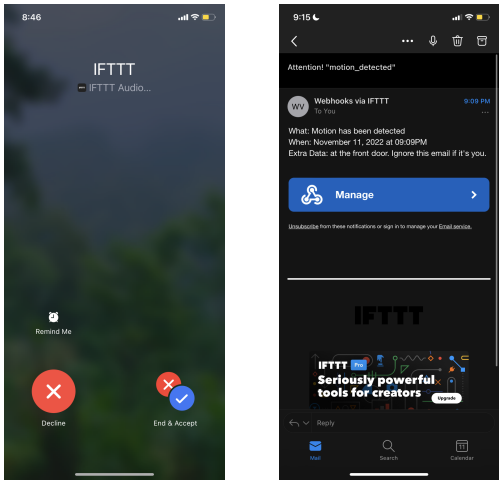


Fig. 11: Printing output whenever a motion is being detected by the PIR Sensor

B. Notifications to the Users

The user receives a call in real-time whenever a motion is detected in the area where the alarm system is installed and also an email with the relevant information as shown in Fig. 12 (a) and Fig. 12 (b), respectively.



(a) Call in real-time (b) Email notification

Fig. 12: Email and Call Notifications

V. Future Work

As a future extension of the work, a camera could be installed with the alarm system, and when motion is detected, it can trigger the camera to activate and record and take pictures of the intruder. Due to financial constraints, we could not incorporate this feature into our project.

VI. Conclusion

The application of a wireless motion detection sensor system on the Arduino platform is proposed in this project using Arduino MKR Wi-Fi 1010 board and a PIR sensor. PIR sensor detects motion in the premises where the alarm system is installed and reports to the user by triggering a call and an email notification immediately. This helps the user to take appropriate action. It's cheaper than other options and easy to use with webhooks integration in Arduino IoT Cloud.

References

[1] Q. I. Sarhan, "Arduino based smart home warning system," in *2020 IEEE 6th International Conference on Control Science and Systems Engineering (ICCSSE)*. IEEE, 2020, pp. 201–206.

[2] S. S. S. M. Soleh, M. M. Som, M. H. Abd Wahab, A. Mustapha, N. A. Othman, and M. Z. Saringat, "Arduino-based wireless motion detecting system," in *2018 IEEE Conference on Open Systems (ICOS)*. IEEE, 2018, pp. 71–75.

[3] K. C. Sahoo and U. C. Pati, "Iot based intrusion detection system using pir sensor," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. IEEE, 2017, pp. 1641–1645.

[4] Arduino. [Online]. Available: https://docs.arduino.cc/hardware/mkr-wifi-1010

[5] Botland. [Online]. Available: https://botland.store/blog/arduino-with-samd21-which-board-to-use/

[6] Atmel-Sparkfun. [Online]. Available: https://cdn.sparkfun.com/datasheets/Dev/Arduino/Boards/Atmel-42181-SAM-D21_Datasheet.pdf

[7] D. Cotriss. [Online]. Available: https://www.businessnewsdaily.com/4919-ifttt-for-business.html

APPENDIX

A. *Code in Web editor of the Arduino IoT Cloud*

Arduino-MKR-WiFi-1010 | ReadMe.adoc | thingProperties.h | Secret

```
/*Copyright@Pranav Jha Student ID 40081750;
                Krupa Sunilbhai Vora Student ID: 40171212,
                Fatemeh Eslaminasab, Student ID: 40170674
  Sketch generated by the Arduino IoT Cloud Thing "PIR Motion Sensor"
  https://create.arduino.cc/cloud/things/eceb87ff-5f74-402c-9d74-4b5af08eae18

  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are
  made to the Thing

  bool motion;


*/
#include "thingProperties.h"

int ledPin = 4;
int inputPin = 5;
int val = 0;


void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if
  //none is found
  delay(1500);

  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  /*
     The following function allows to obtain more information
     related to the state of network and IoT Cloud connection and errors
     the higher number means the more granular information
     The default is 0 (only errors).
     Maximum is 4
  */
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();

  pinMode(ledPin, OUTPUT);
  pinMode(inputPin, INPUT);

}

void loop() {
  ArduinoCloud.update();
  //Your code here

  val = digitalRead(inputPin); // reading input value
  if (val == HIGH) {          // checking if the input is high or not
    digitalWrite(ledPin, HIGH); // if high then turning the led ON
    if (motion == LOW) {
      Serial.println("Motion is Detected in your home!"); //printing the change
      //in output
      motion = HIGH;
    }
  }
  else {
    digitalWrite(ledPin, LOW); \\ turning the LED OFF
    if (motion == HIGH) {
      Serial.println("Motion has Ended!"); //printing the chnage in output
      motion = LOW;
    }
  }

}
```