

# SMART ROOM



Smart Choices  
Smart Lifestyle

MDT 112 Computer Programming For Media

1

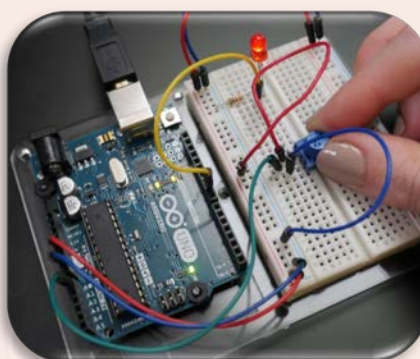
# SOURCE & SIGNIFICANCE



# แหล่งที่มาและความสำคัญ

ปัจจุบัน เทคโนโลยีได้ถูกนำมาประยุกต์ใช้เพื่อให้เกิดความสะดวกสบายตรงตามความต้องการมนุษย์มากยิ่งขึ้น รวมไปถึงการที่นำเทคโนโลยีมาปรับใช้ทางด้านการรักษาความปลอดภัยและความสะดวกสบายในการใช้งานเครื่องใช้ไฟฟ้าภายในบ้านอีกด้วย

เราจึงจำเป็นที่จะต้องนำความรู้และทักษะที่ได้มาจากการเรียนการสอนวิชา MDT 112 ทั้งการเขียนโปรแกรม Arduino สร้างฮาร์ดแวร์ รวมไปถึงวิธีการจัดการ เราได้นำสิ่งเหล่านี้มาวิเคราะห์ ว่าจะสามารถนำความรู้ที่ได้นั้นมาประยุกต์และทำให้เกิดประโยชน์กับชีวิตประจำวันของเราได้มากน้อยเพียงใด สิ่งไหนที่จำเป็นต่อการแก้ไขและสามารถปรับปรุงเพิ่มเติมให้ดีขึ้นได้อย่างไร เพื่อให้ตรงกับความต้องการในชีวิตประจำวันของเรา



ในที่นี้ เราจึงได้สร้างฮาร์ดแวร์ขึ้นมา 3 ชนิด โดยนำมารวมเป็น 1 โครงการ ใช้ชื่อโครงการว่า “ smart Room ” จากการใช้ Arduino และ การต่อวงจร มาประยุกต์ โดยมี 2 ส่วนดังนี้

### Outdoor

- ส่วนหน้าประตู
- ส่วนสแกนบัตรผ่านหน้าประตู



### Indoor

- ส่วนเซนเซอร์ไฟโดยการตรวจจับความเคลื่อนไหวภายในห้อง





2

TARGET  
GROUP



กลุ่มเป้าหมายผู้ใช้งานของเราคือ **ทุกคน**



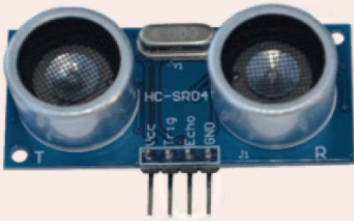
► ทุก ๆ คนสามารถใช้งานฟังก์ชันต่าง ๆ นี้ได้  
เนื่องจากโปรเจกของเราได้คำนึงถึงความต้องการใช้งาน  
และความสะดวกสบายแก่ผู้คนที่ทุกเพศทุกวัย

3

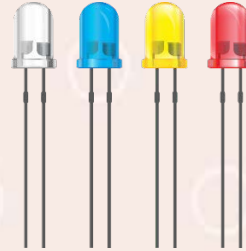
# DESIGN PROCESS



# Main Sensor



Ultrasonic



LED



Buzzer



Motion



NFC / RFID



Button Pin



# CODE

NCF / RFID

RFID\_FINISH

```
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.
int led1 = 6; //red
int led2 = 7; //blue
int buzzer = 3;

void setup()
{
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    pinMode(buzzer, OUTPUT);
    Serial.begin(9600); // Initiate a serial communication
    SPI.begin(); // Initiate SPI bus
    mfrc522.PCD_Init(); // Initiate MFRC522
    Serial.println("Put your card to the reader...");
    Serial.println();
}

void loop()
{
    // Look for new cards
    if ( ! mfrc522.PICC_IsNewCardPresent() )
    {
        return;
    }
    // Select one of the cards
    if ( ! mfrc522.PICC_ReadCardSerial() )
    {
        return;
    }
    //Show UID on serial monitor
    Serial.print("UID tag :");
    String content= "";
    byte letter;
    for (byte i = 0; i < mfrc522.uid.size; i++)
    {
        Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
        Serial.print(mfrc522.uid.uidByte[i], HEX);
    }
}
```

```

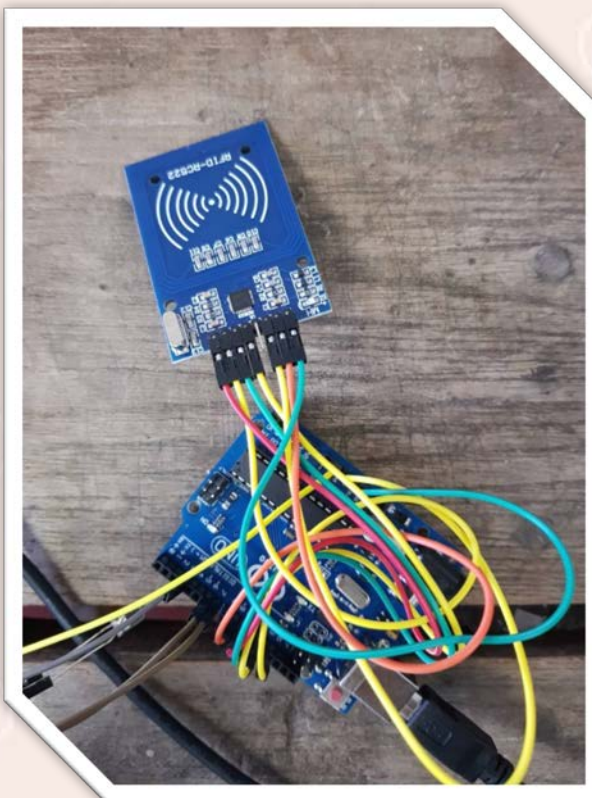
        Serial.print(mfrc522.uid.uidByte[i], HEX);
        content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
        content.concat(String(mfrc522.uid.uidByte[i], HEX));
    }

    Serial.println();
    Serial.print("Message : ");
    content.toUpperCase();
    if (content.substring(1) == "7B DC 4C 7F" || content.substring(1) == "C6 D7 4E 73" || content.substring(1) == "63 5B 60 2E" || content.substring(1)
    || content.substring(1) == "8B B8 55 7F")
    {
        Serial.println("Authorized access");
        Serial.println();

        digitalWrite(led1, LOW);
        digitalWrite(led2, HIGH);
        digitalWrite(buzzer, HIGH);
        delay(200);
        digitalWrite(buzzer, LOW);
        delay(200);
        digitalWrite(buzzer, HIGH);

        digitalWrite(led2, HIGH);
        digitalWrite(buzzer, HIGH);
        delay(200);
        digitalWrite(buzzer, LOW);
        delay(200);
        digitalWrite(buzzer, HIGH);
        delay(500);
        digitalWrite(led1, LOW);
        digitalWrite(led2, LOW);
        digitalWrite(buzzer, LOW);
    }
    else {
        digitalWrite(led1, HIGH);
        digitalWrite(led2, LOW);
        digitalWrite(buzzer, HIGH);
        Serial.println(" Access denied");
        delay(1500);
        digitalWrite(led1, LOW);
        digitalWrite(led2, LOW);
        digitalWrite(buzzer, LOW);
    }
}

```





# CODE

## Motion Sensor

```
motion
int calibrationTime = 10;

//the time when the sensor outputs a low impulse
long unsigned int lowIn;

//the amount of milliseconds the sensor has to be low
//before we assume all motion has stopped
long unsigned int pause = 5000;

boolean lockLow = true;
boolean takeLowTime;

int pirPin = 7;    //the digital pin connected to the PIR sensor's output
int ledPin = 13;

void setup(){
  Serial.begin(9600);
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(pirPin, LOW);

  //give the sensor some time to calibrate
  Serial.print("calibrating sensor ");
  for(int i = 0; i < calibrationTime; i++){
    Serial.print(".");
    delay(1000);
  }
  Serial.println(" done");
  Serial.println("SENSOR ACTIVE");
  delay(5);
}

void loop(){

  if(digitalRead(pirPin) == HIGH){
    digitalWrite(ledPin, HIGH);    //the led visualizes the sensors output pin state
    if(lockLow){
      //makes sure we wait for a transition to LOW before any further output is made:
      lockLow = false;
      Serial.println("---");
      Serial.print("motion detected at ");
      Serial.print(millis()/1000);
      Serial.println(" sec");
      delay(500);
    }
  }
}
```

```

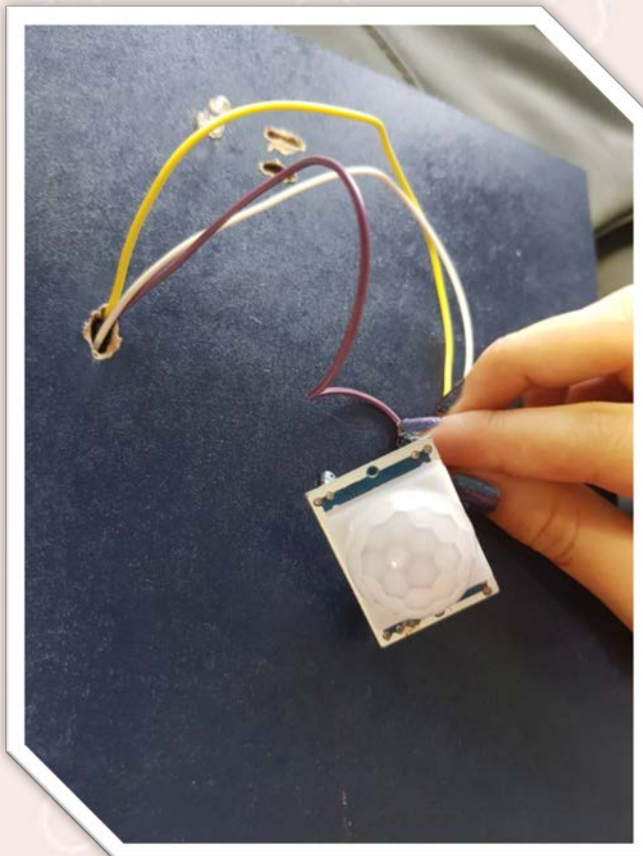
    Serial.println(" sec");
    delay(50);
  }
  takeLowTime = true;
}

if(digitalRead(pirPin) == LOW){
  digitalWrite(ledPin, LOW); //the led visualizes the sensors output pin state

  if(takeLowTime){
    lowIn = millis();          //save the time of the transition from high to LOW
    takeLowTime = false;       //make sure this is only done at the start of a LOW phase
  }

  //if the sensor is low for more than the given pause,
  //we assume that no more motion is going to happen
  if(!lockLow && millis() - lowIn > pause){
    //makes sure this block of code is only executed again after
    //a new motion sequence has been detected
    lockLow = true;
    Serial.print("motion ended at ");          //output
    Serial.print((millis() - pause)/1000);
    Serial.println(" sec");
    delay(50);
  }
}
}

```





# CODE

## Ultrasonic

```
ultrasonic

const int trigPin = 6;
const int echoPin = 7;
const int buzzer = 4;
const int ledPin = 2;

long duration;
int distance;
int safetyDistance;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculating the distance
  distance= duration*0.034/2;

  safetyDistance = distance;
  if (safetyDistance <= 17){
    digitalWrite(buzzer, HIGH);
    digitalWrite(ledPin, HIGH);
  }
  else{
    digitalWrite(buzzer, LOW);
    digitalWrite(ledPin, LOW);
  }

  // Prints the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.println(distance);
}
```



# STRUCTURE

Ultrasonic



LED

Buzzer

NFC / RFID



# STRUCTURE

Motion

LED

Button



4

HOW  
TO  
USE

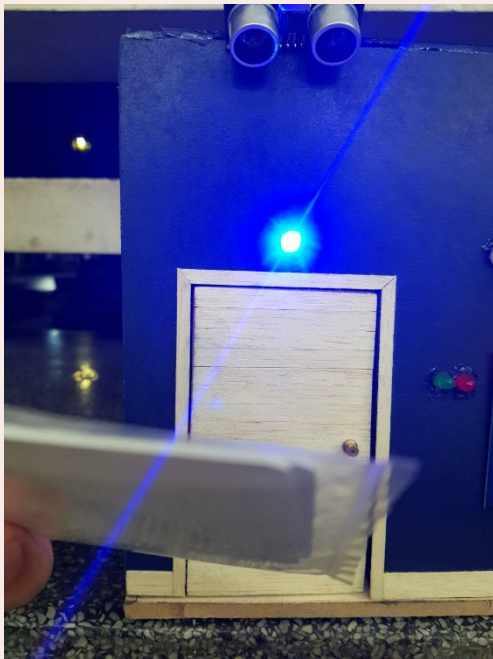




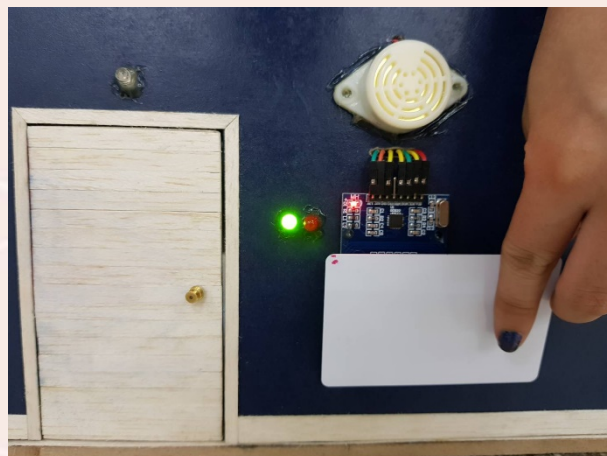
# LET'S START ...



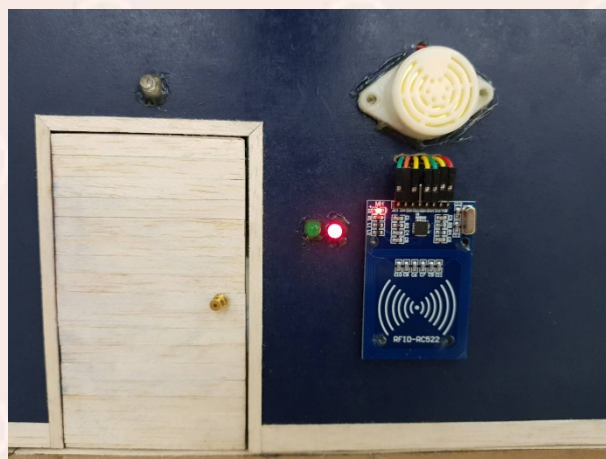
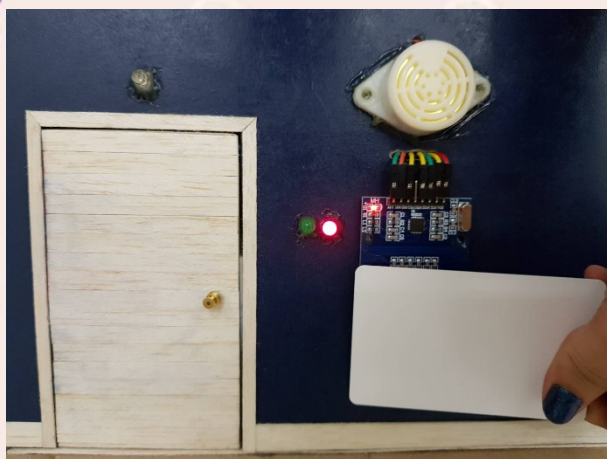
1. เริ่มการใช้งานโดยการตรวจจับความเคลื่อนไหวหน้าประตู เช่น ตรวจจับคนที่อยู่หน้าประตู



2. จากนั้นเซนเซอร์ Ultrasonic จะตรวจจับ จากนั้นจะส่งเสียงพร้อมไฟแจ้งเตือน ให้ไปสแกนบัตรที่เครื่องสแกนบัตร



3. พอสแกนบัตรที่เครื่องสแกนบัตร ถ้ารหัสตรงกับที่เครื่องสแกนต้องการ จะมีเสียงแจ้งเตือนเป็นจังหวะสั้นๆ และขึ้นไฟสีเขียว เครื่องใช้ไฟฟ้าในบ้านจะเปิดพร้อมใช้ทันที

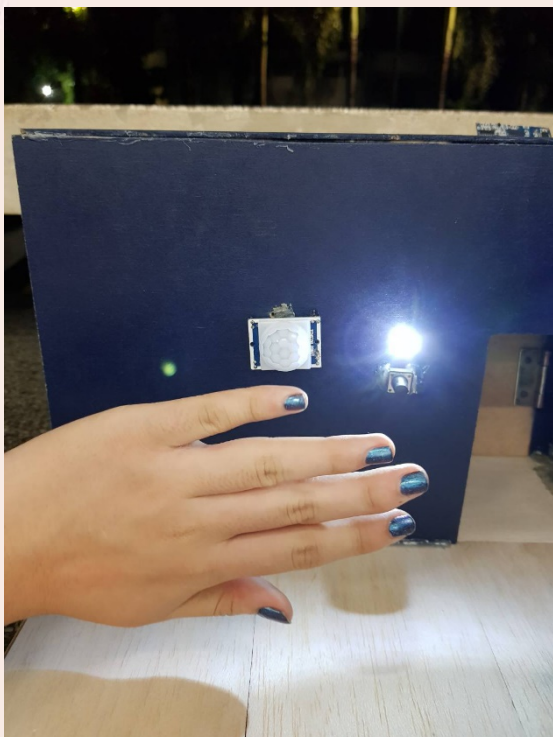


4. แต่ถ้ารหัสไม่ตรงกับที่เครื่องสแกนต้องการ จะมีเสียงแจ้งเตือนเป็นจังหวะยาว และขึ้นไฟสีแดง





5. ตัวเซนเซอร์ Motion ที่พร้อม  
ใช้งานก็จะรอให้เราเดินเข้าไปในห้อง



6. เมื่อมีความเคลื่อนไหวภายใน  
ห้อง เซนเซอร์ตรวจจับความร้อน  
จะทำงานแล้วไฟก็จะติด

