TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES

Ayala Blvd, Ermita, Manila, 1000 Metro Manila

School Year 2019 – 2020

Home Security System

Using Arduino Uno, PIR Motion Sensor, GSM SIM800L Module, 12DC Solenoid Door Lock, and Magnetic Switch Sensor

Submitted to:

Sir Darwin Vargas
Microcontroller Systems

Submitted by:

Alcaraz, Arvie
Angulo, Enna
Arambulo, Franco
Condeza, Ancess
Cortez, Jan Paolo
Danseco, Adhriel
Estanislao, Neziah Hiel
Fabiosa, Julie-Ann
Floresca, Irene Aubrey
Lozano, Jonathan Kyle
Rebadajo, Ericka
Salvador, Jefferson

BSIT – 2A

TABLE OF CONTENTS

I. SYSTEM

Home Security Alarm Systems are very important in present day society, where crime is increasing. With the technological advancements we have achieved in the recent years, a homeowner doesn't have to worry about home security while getting off his/her home. Home Security Systems are an important feature of modern residential and office setups. Home security systems must be affordable, reliable, and effective.

This is a Home Security System using Arduino, which will trigger alarm when a window is opened, or movement detected in the room when the system is activated. The most basic definition of any security system is found in its name. It is literally a means or method by which something is secured through a system of interworking components and devices. In this instance, it is about home security systems, which are networks of integrated electronic devices working together with a central control panel to protect against burglars and other potential home intruders.

This project is designed using normally closed reed switches connected to windows and additional passive infrared (PIR) motion sensors to detect movement of a burglar or an unwanted intruder in your home. The security system can send text message to the number of the user.

This system is a basic motion activated alarm. It is built around an Arduino Microcontroller. It is connected to a PIR motion sensor, a buzzer, a resistor, and a pair of external terminals. The whole system is battery powered so that it is easily portable. Once you have the code, you can connect all the external parts. The easiest way to do this is with a breadboard. This will let you make temporary connections to test everything out.

Our project is a basic motion-sensing alarm that detects when someone enters the area. When an intruder is detected, it activates a siren which is to inform the user in case movement was detected or trigger the buzzer in case window was opened. Our body generates heat energy in the form of infrared which is invisible to human eyes. But it can be detected by electronic sensor. The aim of this project is to implement a simple and affordable, but efficient home security alarm system. The project is designed for detecting intruders and informing the owner by making a text message.

Our project can be deciphered into three main functions. First function is whenever the Passive Infrared (PIR) motion sensor detected movement inside the house it will immediately alert the system, then the system (GSM SIM 800L Module) will send a text message containing “Intruder Alert!” to the user’s number. Second function is when the text message reached the user’s phone number, He / She can reply to the system with the codeword “Unlock”, then when the user text the system, the system will eventually open the door for 20 seconds so the person waiting outside or inside the house can come out or

come inside the house. After 20 seconds of opening the door it will be locked again by the system. Third function is when the window is opened by another person, the alarm will be automatically triggered and then the buzzer will be activated.

In the first function the main property is the Passive Infrared (PIR) motion sensor and the GSMSIM800L Module. The PIR motion sensor will detect the movement inside/ outside the house, while the GSMSIM800L Module will send the text message to the user and it will receive the text whenever the user replied “Unlock” to the system.

In the second function the equipment that we used is the 12DC Solenoid Door Lock and the 5v Relay and of course the GSM SIM800L Module. The use of the 12DC Solenoid Door Lock it will be the doors lock and it will be controlled by the 5v relay, the relay will manage the lock by turning it off and on. But to make this two components work it will need the help of the system, whenever the user send the codeword Unlock to the system, the relay will be put into high state where it will deliver voltage to the solenoid and the door will be unlock for 20 seconds.

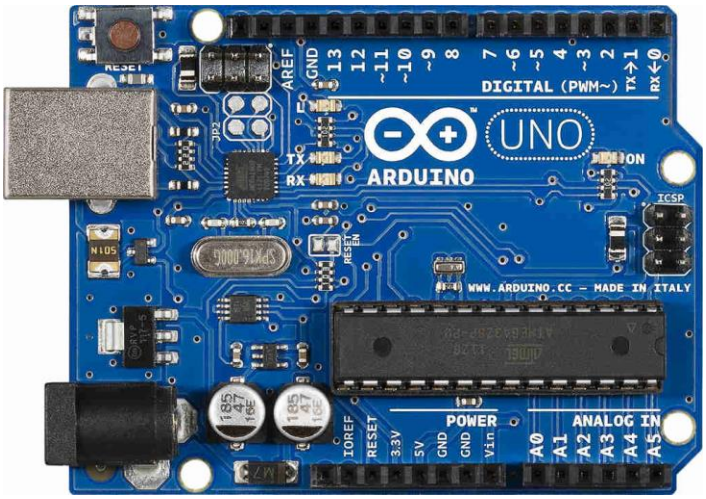
In the third function we use Magnetic switch sensor and buzzer, the magnetic switch is composed of two magnets whenever the two magnets do not touch each other the buzzer will activate but if it touches each magnet the buzzer will not be activated. But when the window is opened the buzzer will also be activated to create an alarming sound.

II. INSTRUCTIONS

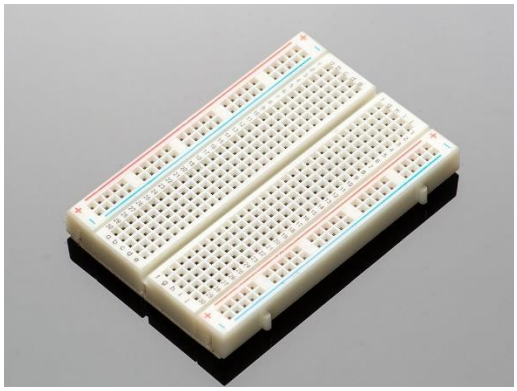
MATERIALS NEEDED

- Arduino IDE (Install ADAFRUIT FONA LIBRARY)

- Arduino UNO



- Half Bread Board



- Solenoid Door Lock

- Magnetic Switch Sensor

- PIR Sensor

- GSM SIM800L

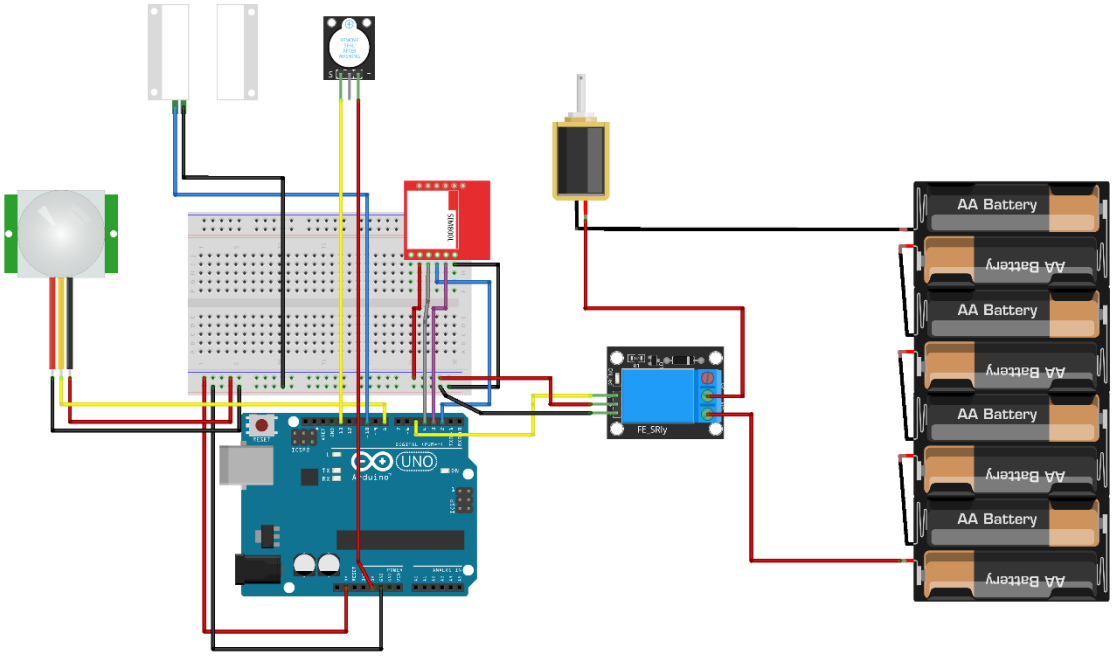
- 3-PIN Active Buzzer

- Jumper Wires

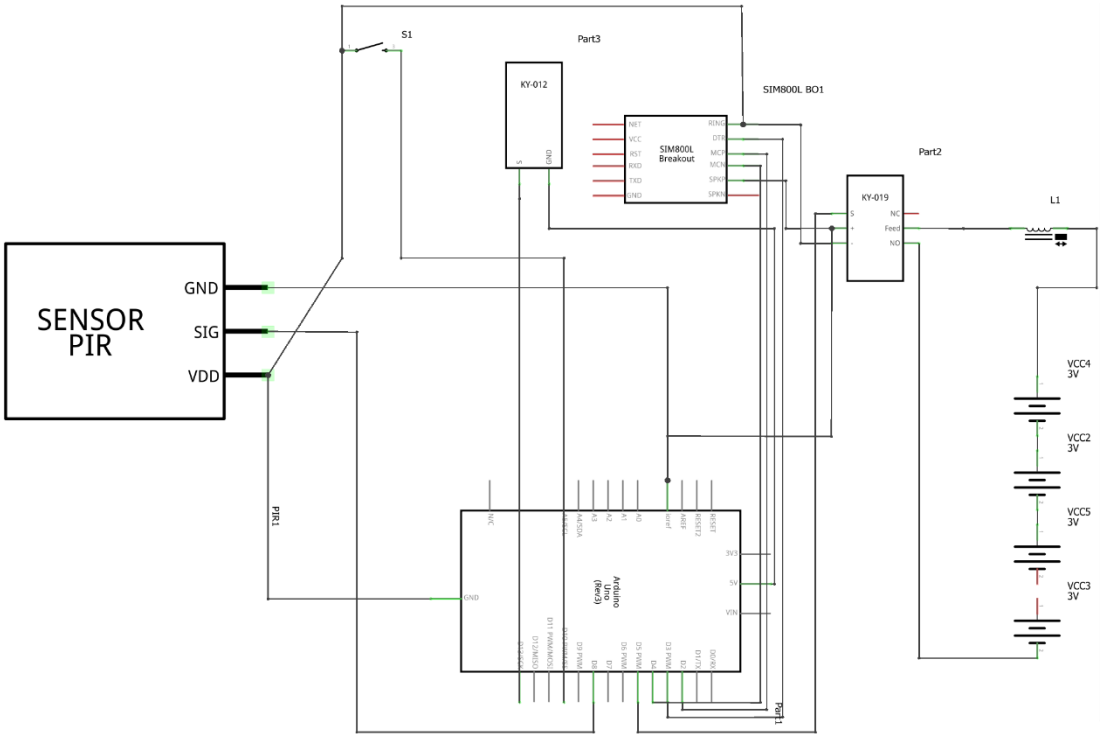
- 5V Relay

- 12V Power Supply

Diagram



Schematic



CODES

systemTry | Arduino 1.8.13 (Windows Store 1.8.42.0)

File Edit Sketch Tools Help

systemTry

```
#include "Adafruit_FONA.h"

#define SMS_CHECK 4000
#define MAGNETIC_CHECK 2000
#define FONA_RX 2
#define FONA_TX 3
#define FONA_RST 4
#define PIR_PIN 8
#define RELAY_PIN 5
//#define LED_RED1 12
//#define LED_RED2 13
#define BUZZER 13
#define MAGNETIC_PIN 10


#include <SoftwareSerial.h>
SoftwareSerial fonaSS = SoftwareSerial(FONA_TX, FONA_RX);
SoftwareSerial *fonaSerial = &fonaSS;

Adafruit_FONA fona = Adafruit_FONA(FONA_RST);

char SMSbuffer[200];
uint16_t SMSLength;
String SMSString = "";

uint8_t type;

uint32_t prev_time = 0;
uint32_t current time = 0;
```

8 Arduino Uno

11:25 AM 7/31/2020

systemTry | Arduino 1.8.13 (Windows Store 1.8.42.0)

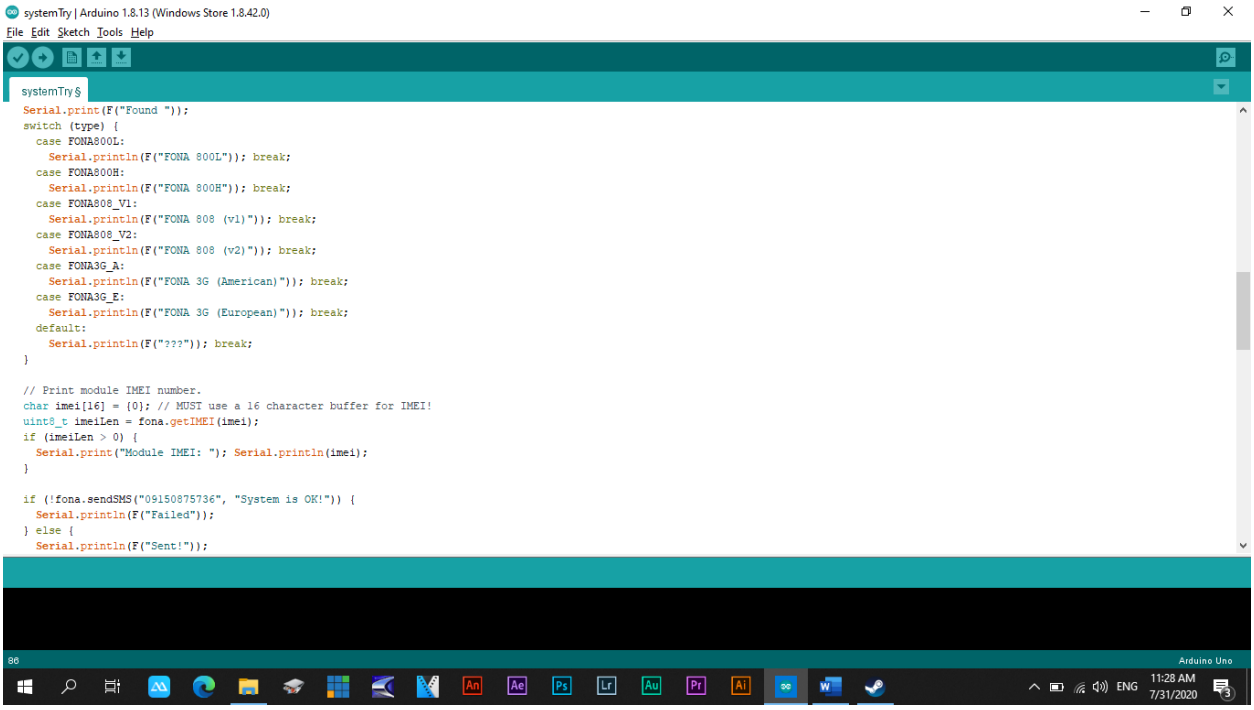
File Edit Sketch Tools Help

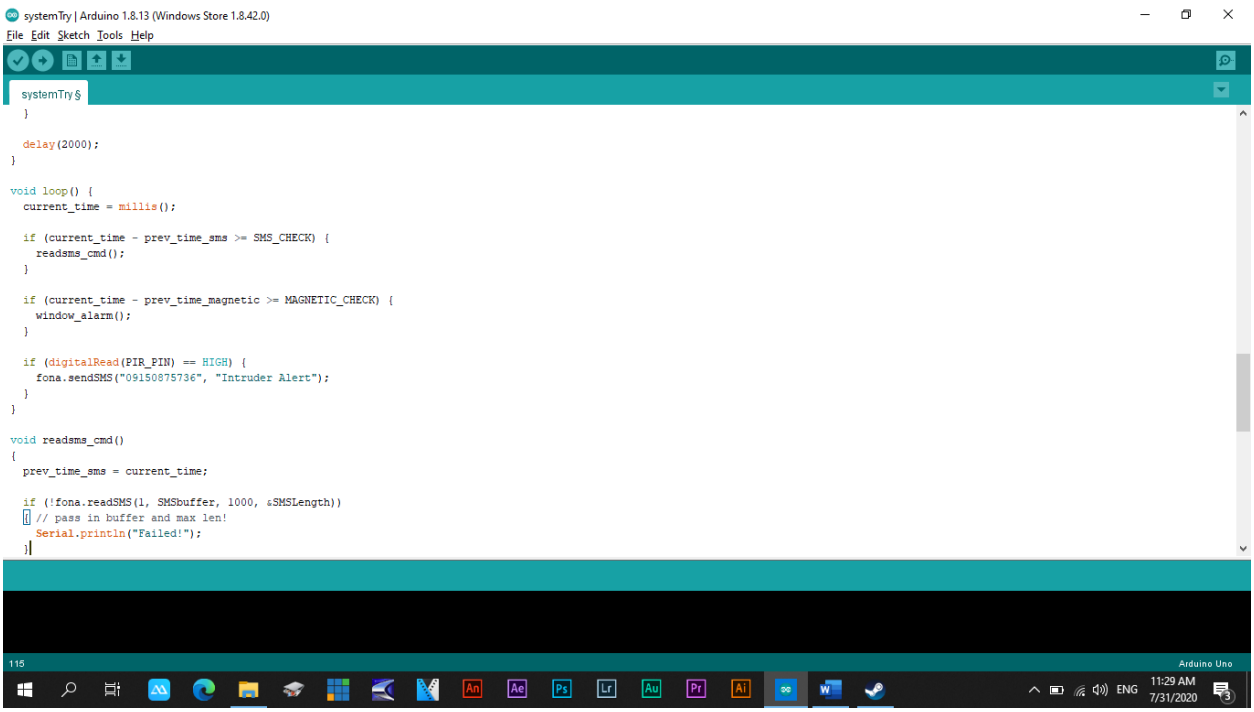
systemTry §

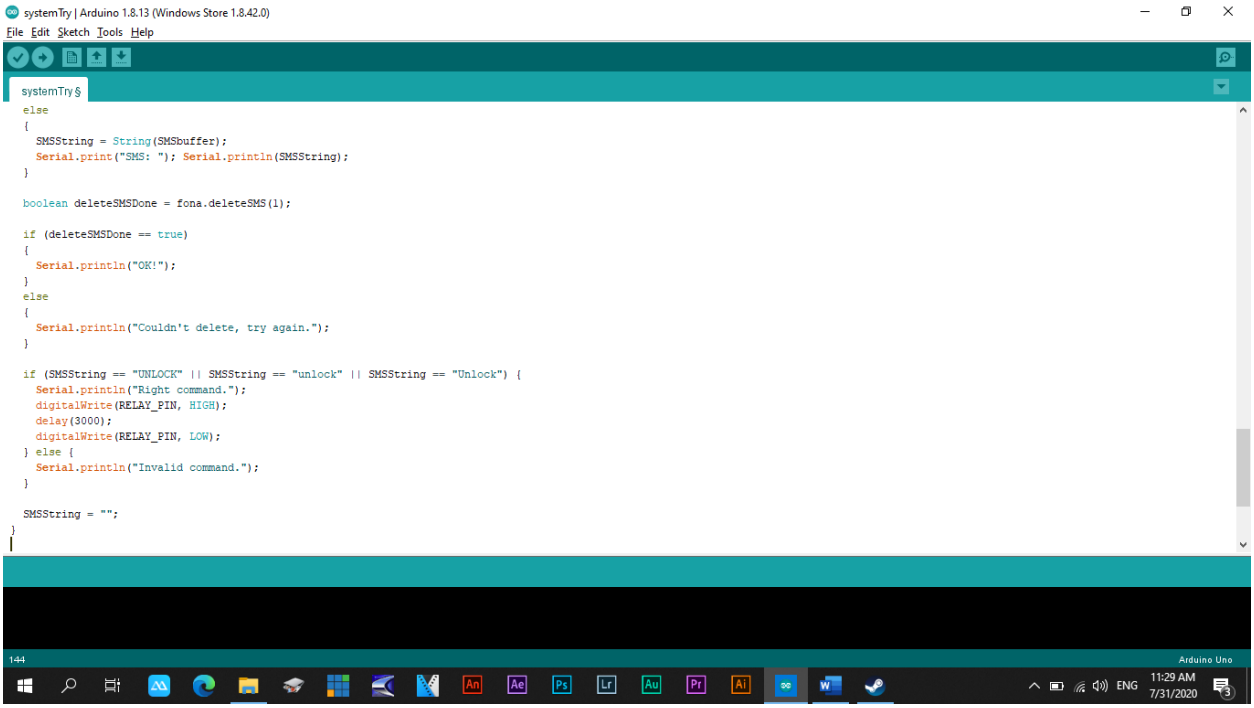
```
uint32_t current_time = 0;
uint32_t prev_time_sms = 0;
uint32_t prev_time_magnetic = 0;

void setup() {
  Serial.begin(115200);

  pinMode(MAGNETIC_PIN, INPUT_PULLUP);
  pinMode(PIR_PIN, INPUT);
  //  pinMode(LED_RED1 , OUTPUT);
  //  pinMode(LED_RED2, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  pinMode(RELAY_PIN, OUTPUT);

  Serial.println(F("Basic Sending"));
  Serial.println(F("Initializing....(May take 3 seconds)"));

  digitalWrite(RELAY_PIN, LOW);
  digitalWrite(BUZZER, HIGH);
  //  digitalWrite(LED_RED1, LOW);
  //  digitalWrite(LED_RED2, LOW);\
  delay(3000);
  fonaSerial->begin(9600);
  if (! fona.begin(*fonaSerial)) {
    Serial.println(F("Couldn't find FONA"));
    while (1);
  }
  type = fona.type();
  Serial.println(F("FONA is OK"));
```

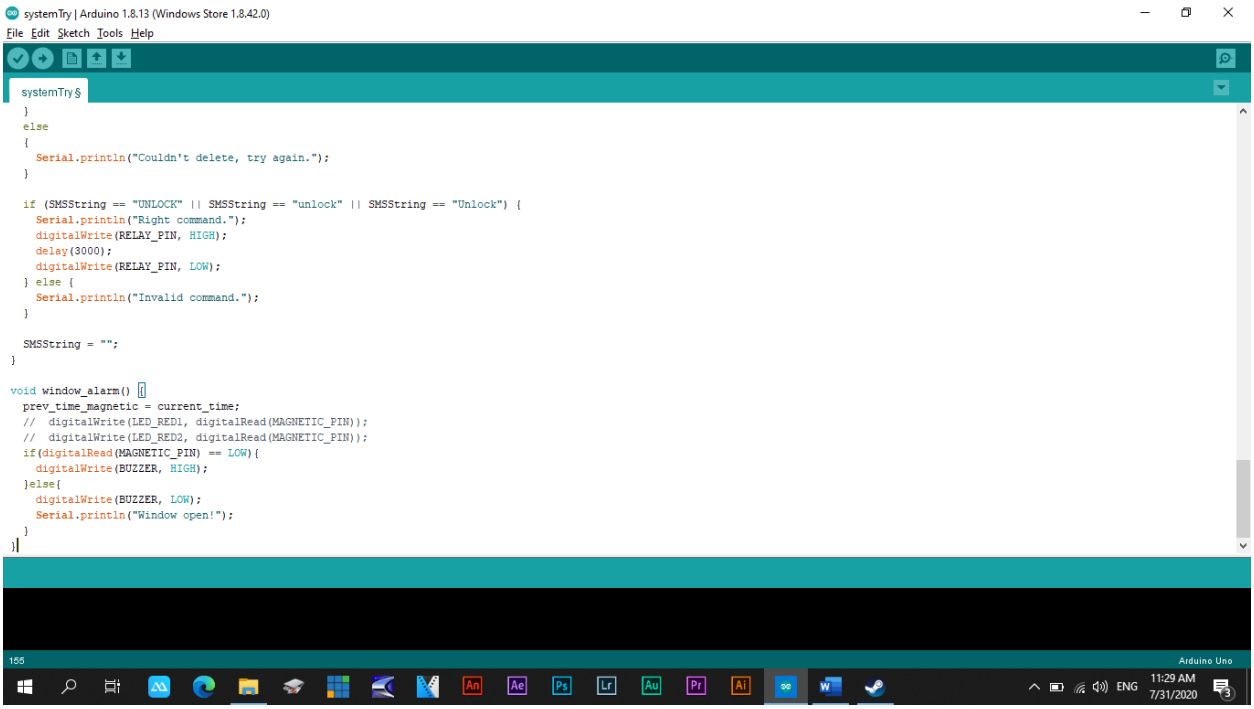
57 Arduino Uno

11:28 AM 7/31/2020


systemTry | Arduino 1.8.13 (Windows Store 1.8.42.0)
File Edit Sketch Tools Help
systemTry §
Serial.print(F("Found "));
switch (type) {
case FONA800L:
Serial.println(F("FONA 800L")); break;
case FONA800H:
Serial.println(F("FONA 800H")); break;
case FONA808_V1:
Serial.println(F("FONA 808 (v1)")); break;
case FONA808_V2:
Serial.println(F("FONA 808 (v2)")); break;
case FONA3G_A:
Serial.println(F("FONA 3G (American)")); break;
case FONA3G_E:
Serial.println(F("FONA 3G (European)")); break;
default:
Serial.println(F("???")); break;
}
// Print module IMEI number.
char imei[16] = {0}; // MUST use a 16 character buffer for IMEI!
uint8_t imeiLen = fona.getIMEI(imei);
if (imeiLen > 0) {
Serial.print("Module IMEI: "); Serial.println(imei);
}
if (!fona.sendSMS("09150875736", "System is OK!")) {
Serial.println(F("Failed"));
} else {
Serial.println(F("Sent!"));
86
Arduino Uno
ENG
11:28 AM
7/31/2020


systemTry | Arduino 1.8.13 (Windows Store 1.8.42.0)
File Edit Sketch Tools Help
systemTry §
}
delay(2000);
}
void loop() {
current_time = millis();
if (current_time - prev_time_sms >= SMS_CHECK) {
readsms_cmd();
}
if (current_time - prev_time_magnetic >= MAGNETIC_CHECK) {
window_alarm();
}
if (digitalRead(PIR_PIN) == HIGH) {
fona.sendSMS("09150875736", "Intruder Alert");
}
}
void readsms_cmd()
{
prev_time_sms = current_time;
if (!fona.readSMS(1, SMSbuffer, 1000, &SMSLength))
{ // pass in buffer and max len!
Serial.println("Failed!");
}
115
Arduino Uno
ENG
11:29 AM
7/31/2020


systemTry | Arduino 1.8.13 (Windows Store 1.8.42.0)
File Edit Sketch Tools Help
systemTry §
else
{
SMSString = String(SMSbuffer);
Serial.print("SMS: "); Serial.println(SMSString);
}
boolean deleteSMSDone = fona.deleteSMS(1);
if (deleteSMSDone == true)
{
Serial.println("OK!");
}
else
{
Serial.println("Couldn't delete, try again.");
}
if (SMSString == "UNLOCK" || SMSString == "unlock" || SMSString == "Unlock") {
Serial.println("Right command.");
digitalWrite(RELAY_PIN, HIGH);
delay(3000);
digitalWrite(RELAY_PIN, LOW);
} else {
Serial.println("Invalid command.");
}
SMSString = "";
}
144
Arduino Uno
ENG
11:29 AM
7/31/2020


systemTry | Arduino 1.8.13 (Windows Store 1.8.42.0)
File Edit Sketch Tools Help
systemTry §
}
else
{
Serial.println("Couldn't delete, try again.");
}
if (SMSString == "UNLOCK" || SMSString == "unlock" || SMSString == "Unlock") {
Serial.println("Right command.");
digitalWrite(RELAY_PIN, HIGH);
delay(3000);
digitalWrite(RELAY_PIN, LOW);
} else {
Serial.println("Invalid command.");
}
SMSString = "";
}
void window_alarm() {
prev_time_magnetic = current_time;
// digitalWrite(LED_RED1, digitalRead(MAGNETIC_PIN));
// digitalWrite(LED_RED2, digitalRead(MAGNETIC_PIN));
if(digitalRead(MAGNETIC_PIN) == LOW){
digitalWrite(BUZZER, HIGH);
}else{
digitalWrite(BUZZER, LOW);
Serial.println("Window open!");
}
}
155
Arduino Uno
ENG
11:29 AM
7/31/2020

```
#include "Adafruit_FONA.h"
#define SMS_CHECK 4000
#define MAGNETIC_CHECK 2000
#define FONA_RX 2
#define FONA_TX 3
#define FONA_RST 4
#define PIR_PIN 8
#define RELAY_PIN 5
//#define LED_RED1 12
//#define LED_RED2 13
#define BUZZER 13
#define MAGNETIC_PIN 10

#include <SoftwareSerial.h>
SoftwareSerial fonaSS =
SoftwareSerial(FONA_TX, FONA_RX);
SoftwareSerial *fonaSerial = &fonaSS;

Adafruit_FONA fona =
Adafruit_FONA(FONA_RST);

char SMSbuffer[200];
uint16_t SMSLength;
String SMSString = "";

uint8_t type;

uint32_t prev_time = 0;
uint32_t current_time = 0;
uint32_t prev_time_sms = 0;
uint32_t prev_time_magnetic = 0;

void setup() {
  Serial.begin(115200);

  pinMode(MAGNETIC_PIN,
INPUT_PULLUP);
  pinMode(PIR_PIN, INPUT);
  //  pinMode(LED_RED1 , OUTPUT);
  //  pinMode(LED_RED2, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  pinMode(RELAY_PIN, OUTPUT);

  Serial.println(F("Basic Sending"));
  Serial.println(F("Initializing....(May take 3
seconds)"));

  digitalWrite(RELAY_PIN, LOW);
  digitalWrite(BUZZER, HIGH);
  //  digitalWrite(LED_RED1, LOW);
  //  digitalWrite(LED_RED2, LOW);\
  delay(3000);
  fonaSerial->begin(9600);
  if (! fona.begin(*fonaSerial)) {
    Serial.println(F("Couldn't find FONA"));
    while (1);
  }
  type = fona.type();
  Serial.println(F("FONA is OK"));
  Serial.print(F("Found "));
  switch (type) {
    case FONA800L:
      Serial.println(F("FONA 800L")); break;
    case FONA800H:
      Serial.println(F("FONA 800H")); break;
    case FONA808_V1:
      Serial.println(F("FONA 808 (v1)")); break;
    case FONA808_V2:
      Serial.println(F("FONA 808 (v2)")); break;
    case FONA3G_A:
      Serial.println(F("FONA 3G (American)"));
break;
    case FONA3G_E:
      Serial.println(F("FONA 3G (European)"));
break;
    default:
      Serial.println(F("???")); break;
  }

  // Print module IMEI number.
  char imei[16] = {0}; // MUST use a 16
character buffer for IMEI!
  uint8_t imeiLen = fona.getIMEI(imei);
  if (imeiLen > 0) {
    Serial.print("Module IMEI: ");
Serial.println(imei);
  }
```

```
  if (!fona.sendSMS("09150875736",
"System is OK!")) {
    Serial.println(F("Failed"));
  } else {
    Serial.println(F("Sent!"));
  }

  delay(2000);
}

void loop() {
  current_time = millis();

  if (current_time - prev_time_sms >=
SMS_CHECK) {
    readsms_cmd();
  }

  if (current_time - prev_time_magnetic >=
MAGNETIC_CHECK) {
    window_alarm();
  }

  if (digitalRead(PIR_PIN) == HIGH) {
    fona.sendSMS("09150875736", "Intruder
Alert");
  }
}

void readsms_cmd()
{
  prev_time_sms = current_time;

  if (!fona.readSMS(1, SMSbuffer, 1000,
&SMSLength))
  { // pass in buffer and max len!
    Serial.println("Failed!");
  }
  else
  {
    SMSString = String(SMSbuffer);
    Serial.print("SMS: ");
Serial.println(SMSString);
  }

  boolean deleteSMSDone =
fona.deleteSMS(1);

  if (deleteSMSDone == true)
  {
    Serial.println("OK!");
  }
  else
  {
    Serial.println("Couldn't delete, try
again.");
  }

  if (SMSString == "UNLOCK" || SMSString
== "unlock" || SMSString == "Unlock") {
    Serial.println("Right command.");
    digitalWrite(RELAY_PIN, HIGH);
    delay(3000);
    digitalWrite(RELAY_PIN, LOW);
  } else {
    Serial.println("Invalid command.");
  }

  SMSString = "";
}

void window_alarm() {
  prev_time_magnetic = current_time;
  //  digitalWrite(LED_RED1,
digitalRead(MAGNETIC_PIN));
  //  digitalWrite(LED_RED2,
digitalRead(MAGNETIC_PIN));
  if(digitalRead(MAGNETIC_PIN) == LOW){
    digitalWrite(BUZZER, HIGH);
  }else{
    digitalWrite(BUZZER, LOW);
    Serial.println("Window open!");
  }
}
```

III. ACKNOWLEDGEMENT

We offer our gratitude for our Professor, Mr. Darwin Vargas for helping us in paving the way so that we can have our components, for his patience in guiding us through the process and helping us complete our project despite the difficulties that this pandemic has brought upon us.

We also offer our deepest gratitude for the Arduino Philippines United Community for their assistance and help that they have offered with the project especially in a dire time like this due to the pandemic. We would like to thank them for providing the materials that we needed such as the GSM SIM800L module, Magnetic Switch Sensor, PIR motion sensor, 3 Pin Active Buzzer, and Solenoid Door Lock, without them this project would not be possible.

We value and look upon this kindness with deep gratitude for with them, none of it would have been successful

IV. REFERENCES

https://github.com/adafruit/Adafruit_F0NA

https://www.eprojectszone.com/sending-and-receiving-messages-with-sim800l-gsm-module-and-arduino/

https://devcraze.com/tutorials/read-and-send-sms-using-sim800l-gsm-module-and-arduino/

https://devcraze.com/tutorials/read-and-send-sms-using-sim800l-gsm-module-and-arduino/

https://randomnerdtutorials.com/arduino-with-pir-motion-sensor/?fbclid=IwAR1j43gCtUoKxQZxO9p3ooL2su5IYTbkeheut_tZbFaxuiMJnqxeaLjXH5s

https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/?fbclid=IwAR3vegqSjRm10gFx02R5QD1NejA-DXYG_gnC-gg4qARx8CI9XX546Vd7CqU

https://www.youtube.com/watch?v=wGU04jtHC9w

https://www.electroschematics.com/sim800l-one-channel-sms-relay/

https://www.youtube.com/watch?v=DfAzbU-nEcs

https://www.tutorialspoint.com/arduino/arduino_pir_sensor.htm

https://www.youtube.com/watch?v=FxaTDvs34mM&t=10s&fbclid=IwAR3CQs8MAL4QF1FFs9JExkMwbBmZLFCPjggNlAtjjNr1IPeGhRM-EohOVIk

https://www.youtube.com/watch?v=cXoJI49ilvk

V. Meetings Documentation