



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

Lecture 34: Introduction to Internet of Things

CONCEPTS COVERED

- ☐ Internet of Things (IoT)
- ☐ IoT and embedded systems
- ☐ A case study: object tracking



Internet of Things (IoT)

- IoT is the network of physical objects (devices, human beings, vehicles, buildings, etc.) embedded with electronics, software, sensors and network connectivity.
 - Enables the objects to collect and exchange data.
- Wide variety of applications:
 - Smart city
 - Healthcare
 - Object tracking
 - Surveillance and security
 - Limited by imagination...



IoT and Embedded Systems

- IoT constitutes the larger picture.
 - Billions of “*things*” generate data and communicate among each other.
- Each IoT node is essentially an embedded system.
 - IoT revolution has given embedded systems a new lease of life.
- Some typical applications leading to embedded system design:
 - Smart home
 - Object tracking
 - Smart transportation
 - Leading to smart cities → smart world

A Case Study

- We consider object tracking as a case study of IoT.
 - Other applications will be very similar.
 - The basic building blocks that would make up the system are the same in most applications.
- The final implementation is essentially an embedded design problem.
 - Use a suitable microcontroller and interface with the required sensors/actuators to build up the system.
 - For many IoT applications, cost can be a major issue.
 - The choice of subsystems may be dictated by cost issues.

Introduction to Object Tracking

- What is object tracking?
 - A mechanism to locate any object in real time.
 - An object can be any conceivable thing that changes its location over time:
 - a) A vehicle (bicycle, car, bus, truck, airplane)
 - b) A human being
 - c) An animal (cow, dog, wild life, etc.)
 - d) A bag
 - e) ... and many more

Why do we need object tracking?

Example 1 :: Delivery of medicines

- It is required to deliver some medicines under some required environmental conditions.
 - a) Hand over the medicines to some delivery agency.
 - b) Along with the medicines, attach some device that will sense the temperature, humidity and other relevant environmental conditions, along with its location.
 - c) The information from the various sensors shall be uploaded to a server periodically through some suitable communication mechanism (e.g. GPRS, SMS).
 - d) The concerned persons can track the consignments in real time, and may also have access to historical data.

Example 2 :: Tracking of living beings

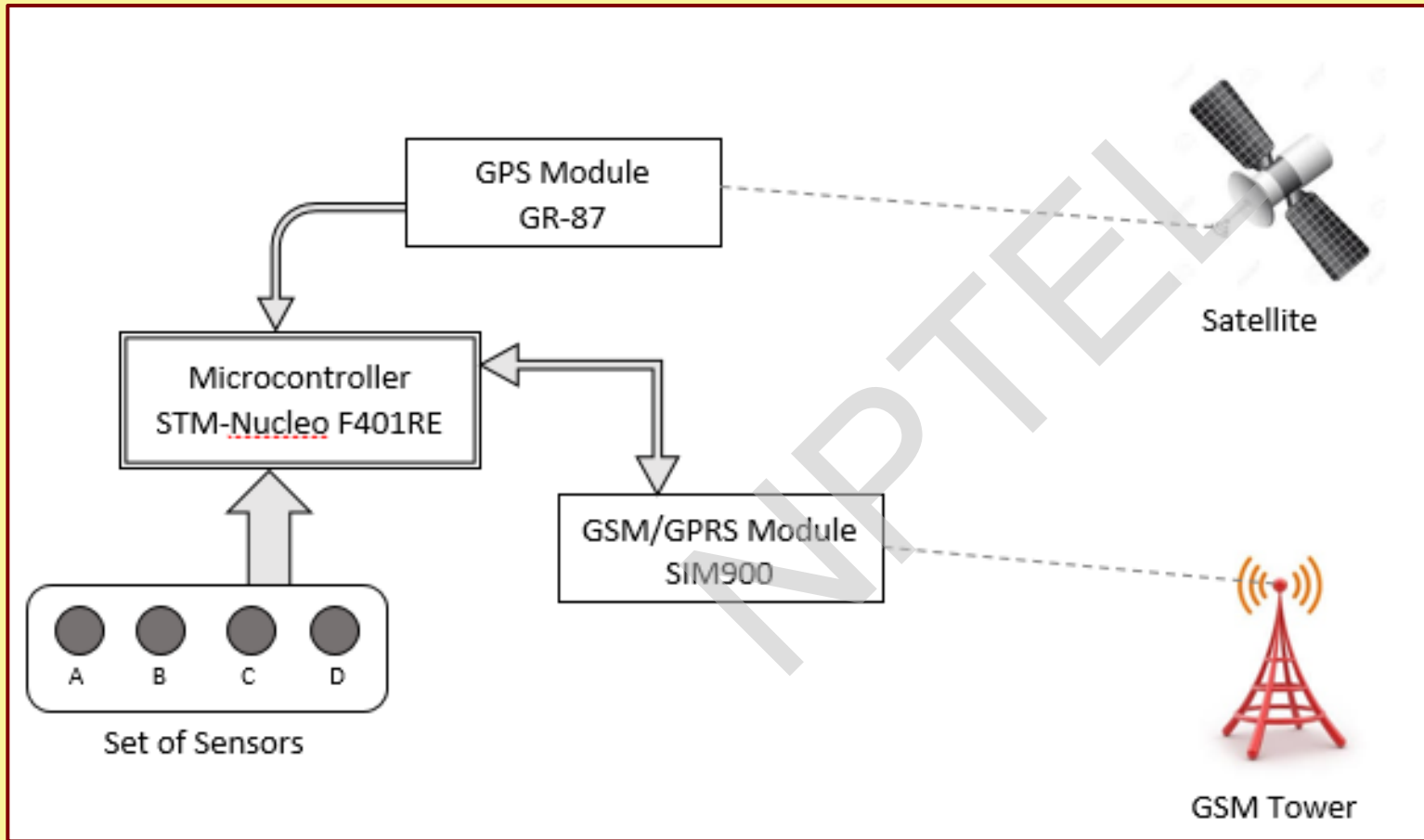
The mechanism is very similar; only the sensors that are required shall vary from one situation to the next.

- a) Tracking children when they are playing outside or playing.
- b) Tracking wild life to gain information about their lifestyle.
- c) Tracking of pets and cattle.

Requirements for building an object tracker

- There are four basic requirements for developing an object tracking system:
 - a) **Processing** :: A microcontroller is required for carrying out some local processing, and interface with the various sensors and the communication subsystems.
 - b) **Communication Capability** :: The system must have some mechanism to communicate with some central server (e.g. GSM/GPRS module).
 - c) **Location Sensing** :: The most important criterion of an object tracker is to accurately locate its position. A GSM module can be used for this purpose.
 - d) **Parameter Sensing** :: Depending on the application, various physical parameters like temperature, humidity, pressure, etc. may need to be sensed. Such sensors must be interfaced with the microcontroller.

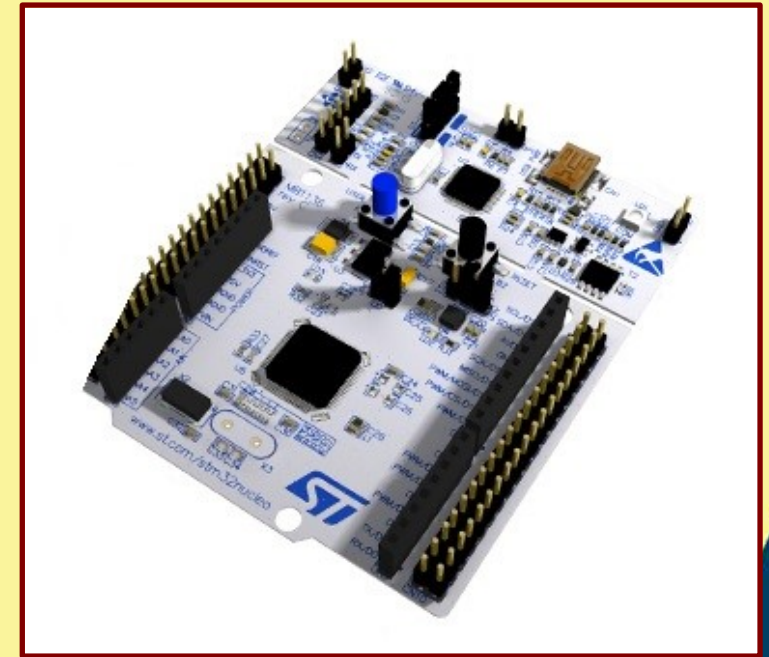
Overall schematic diagram



(a) The microcontroller

Single-chip computer system containing a processor core, memory, and programmable input/output subsystems.

- Responsible for interfacing with the sensors and location tracker, read the parameter data periodically.
- Responsible for interfacing with the communication subsystem, upload data periodically, and respond to incoming commands.
- Example: STM-Nucleo F401RE with ARM Cortex-M4 microprocessor, 96KB SRAM, 512KB flash storage, several I/O ports, etc.



(b) The GPS module

Basically used for location sensing.

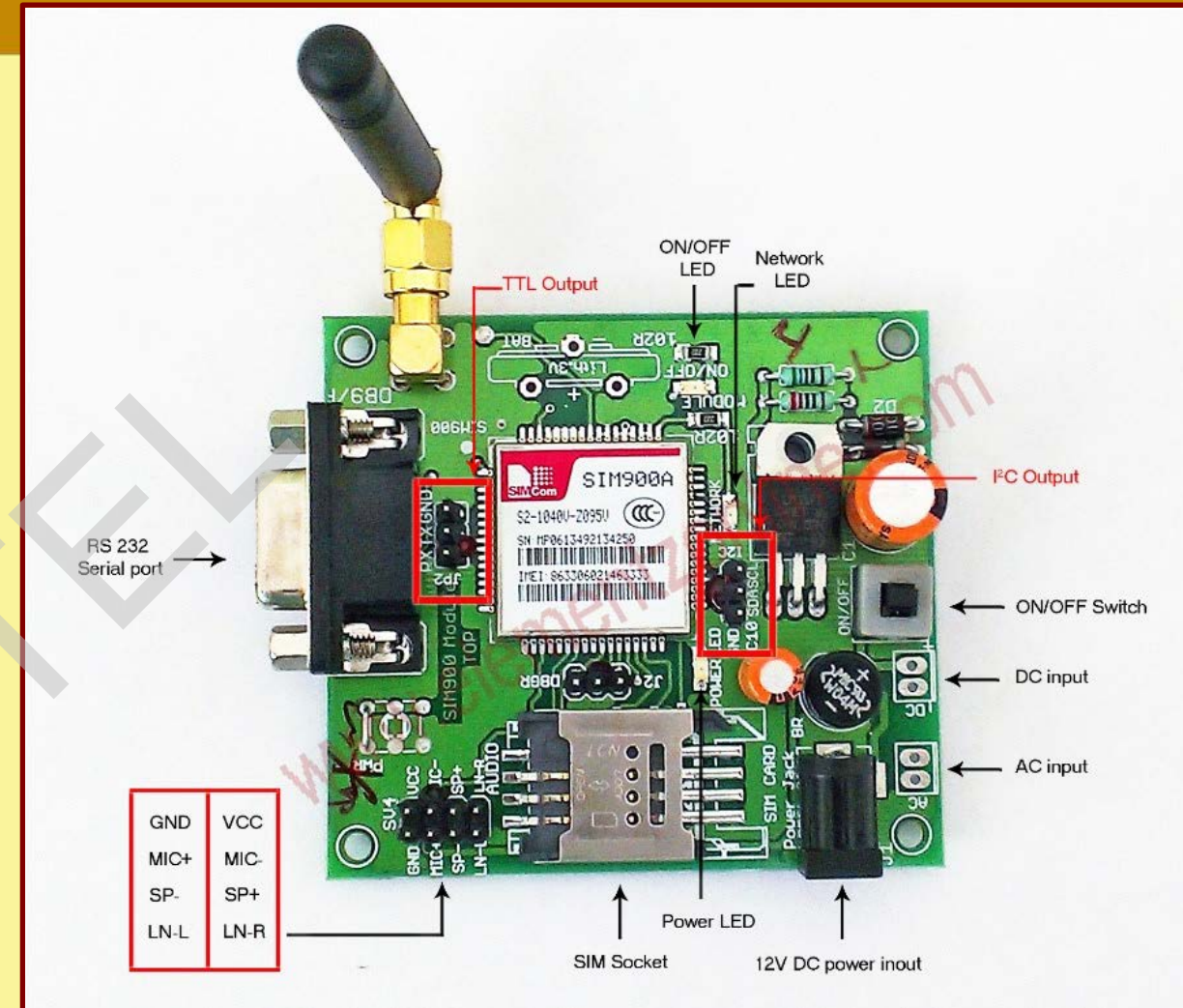
- The Global Positioning System (GPS) is a satellite-based navigation system made up of a network of satellites.
- It can be used anywhere in the world, 24 hours a day.
- Example: Holux GR-87 GPS module, which supports the standard NMEA-0183 format.



(c) The GSM module

Basically used for communication with a remote location.

- Uses the same technology as used by the mobile phone networks for communication.
- It can be used provided there is a mobile tower in the range.
- Example: *SIM900A GSM module*, which supports the standard AT command set. It is a tri-band GSM/GPRS engine that works on frequencies 900 MHz, 1800 MHz, and 1900 MHz.



(d) Sensors

Depends on the application:

- Temperature sensor
- Humidity sensor
- Pressure sensor
- Vibration sensor
- Gas sensor

For communication, other methods like Bluetooth, radio communication, near field communication, etc. can also be used.



NPTEL ONLINE CERTIFICATION COURSES

Thank
you!



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

Lecture 35: GSM and Bluetooth

CONCEPTS COVERED

- ❑ GSM: technology and interfacing
- ❑ Bluetooth: technology and interfacing

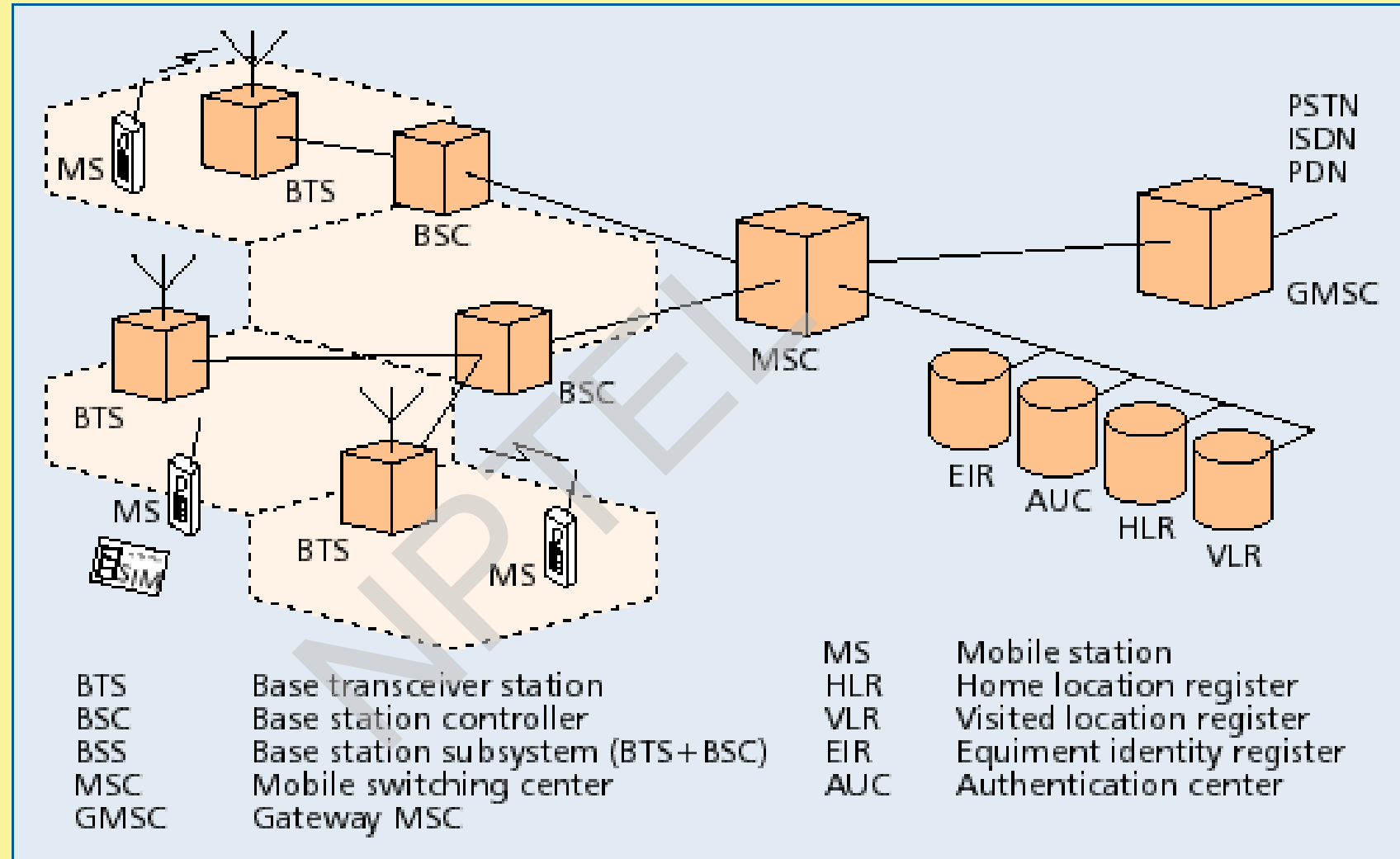


Global System for Mobile (GSM) Communication

What is GSM?

- GSM is a digital mobile network that is widely used by mobile phone users.
- It uses a variation of time division multiple access (TDMA).
 - GSM digitizes and compresses data, then sends it over a channel with two other streams of user data, each in its own time slot.
- GSM, along with other technologies, is part of the evolution of mobile telecommunications, which include:
 - General Packet Radio Service (GPRS)
 - Enhanced Data GSM Environment (EDGE)
 - Universal Mobile Telecommunications Service (UMTS)

GSM System Architecture



GSM Subsystems

- Mobile Station (MS)
 - Consists of two components:
 - a) Mobile Equipment (ME)
 - b) Subscriber Identity Module (SIM)
 - Allows separation of user mobility from equipment mobility.
- Base Transceiver Station (BTS)
 - One per cell; consists of high-speed transmitter and receiver.
 - Provides two channels: signaling and data channel.
- Base Station Controller (BSC)
 - Controls multiple BTS.
 - Performs radio resource management, time and frequency synchronization signals to BTSs.

- Mobile Switching Center (MSC)
 - Switching node of a Public Land Mobile Network (PLMN).
 - Manages allocation of radio resources (handoff), and mobility of subscribers (location registration).
 - There can be several MSCs in a PLMN.
- Gateway MSC (GMSC)
 - Connects mobile network to a fixed network.
- Home Location Register (HLR)
 - For all registered users, it keeps user profile.
 - MSCs exchange information with HLR.
- Visitor Location Register (VLR)
 - Contains temporary information about mobile subscribers that are currently located in the MSC service area but whose HLR are elsewhere.

GSM Identifiers

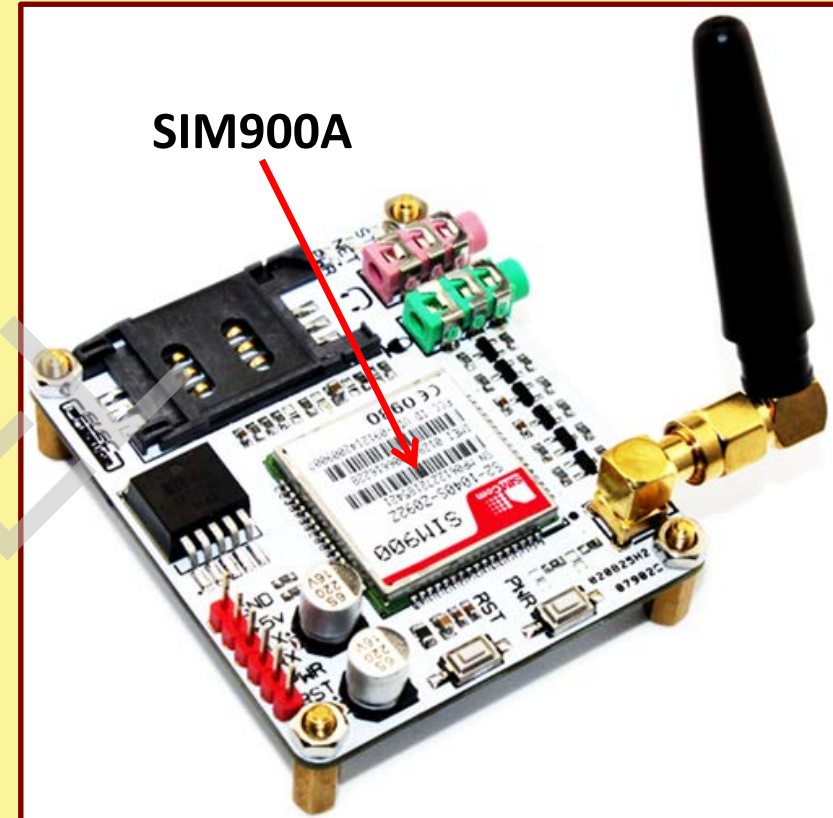
- International mobile subscriber identity (IMSI)
 - Unique 15 digits assigned by service provider, including home country code.
- International mobile station equipment identity (IMEI)
 - Unique 15 digits assigned by equipment manufacturer.
- Temporary mobile subscriber identity (TMSI)
 - 32-bit number assigned by VLR to uniquely identify a mobile station within a VLR's area.

AT Command Set for GSM Modem

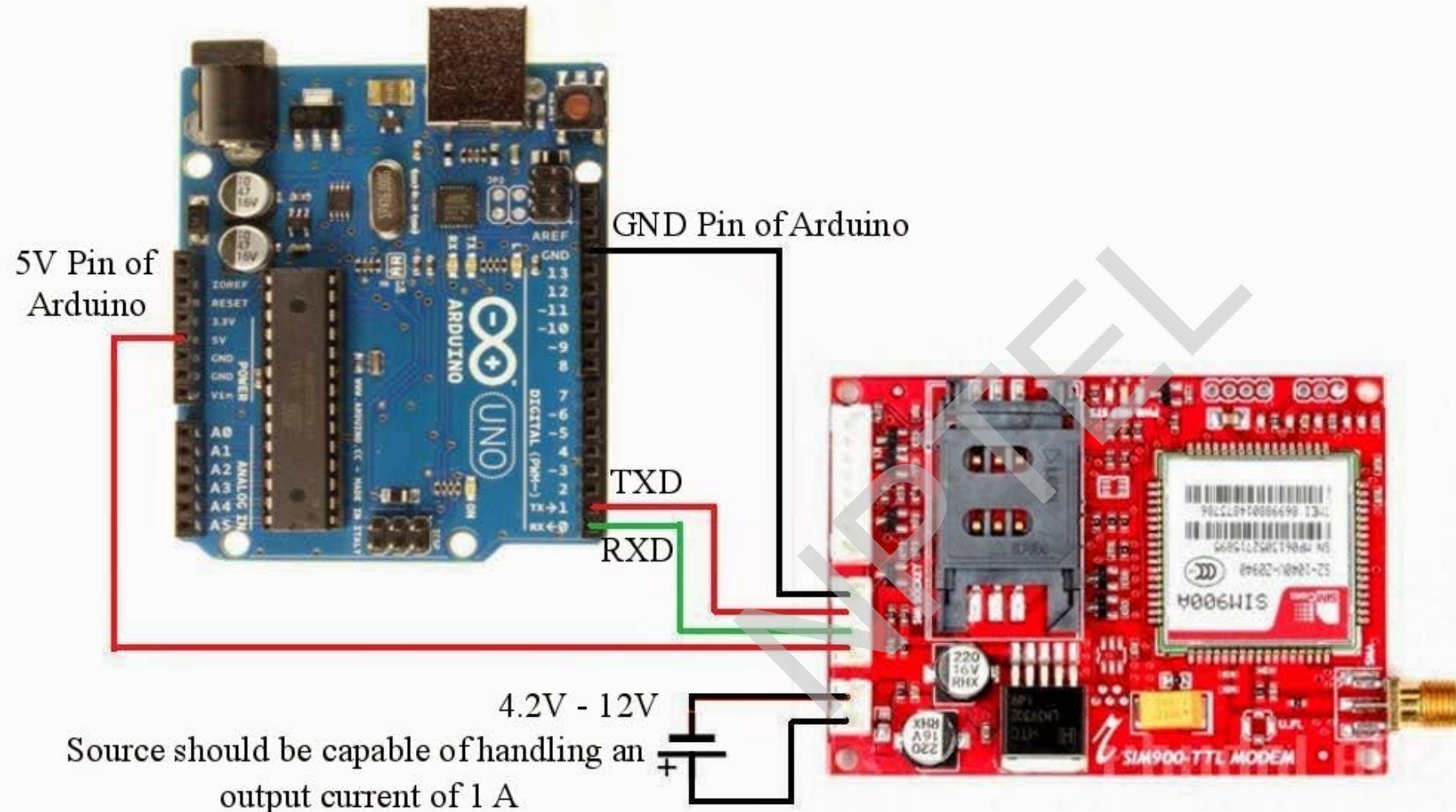
- AT commands are used to control MODEMs.
- AT commands with a GSM/GPRS MODEM can be used to access the following information and services:
 - a) Information and configuration pertaining to MODEM and SIM card.
 - b) SMS service.
 - c) MMS service.
 - d) Fax service.
 - e) Data and voice link over mobile network.

SIM900A GPS/GPRS MODEM

- The SIM900A is a quad-band GPS/GPRS engine, which works on frequencies 850 MHz, 900 MHz, 1800 MHz and 1900 MHz.
- Can be directly interfaced to the serial port of PC.
 - The baud rate can be configured from 9600-115200 through AT commands.
- Connections:
 - Digital I/O pins **RXD** and **TXD** connected to digital port lines.
 - An external power supply through adapter.
 - +5V and GND pins for other circuits.



Typical Interface



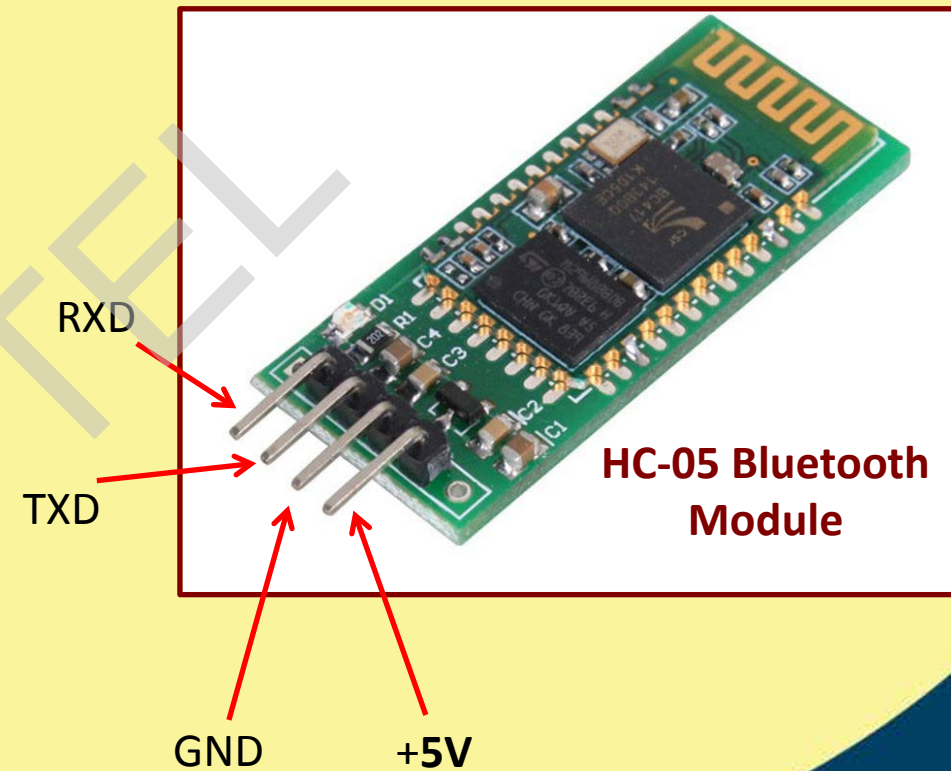
Bluetooth

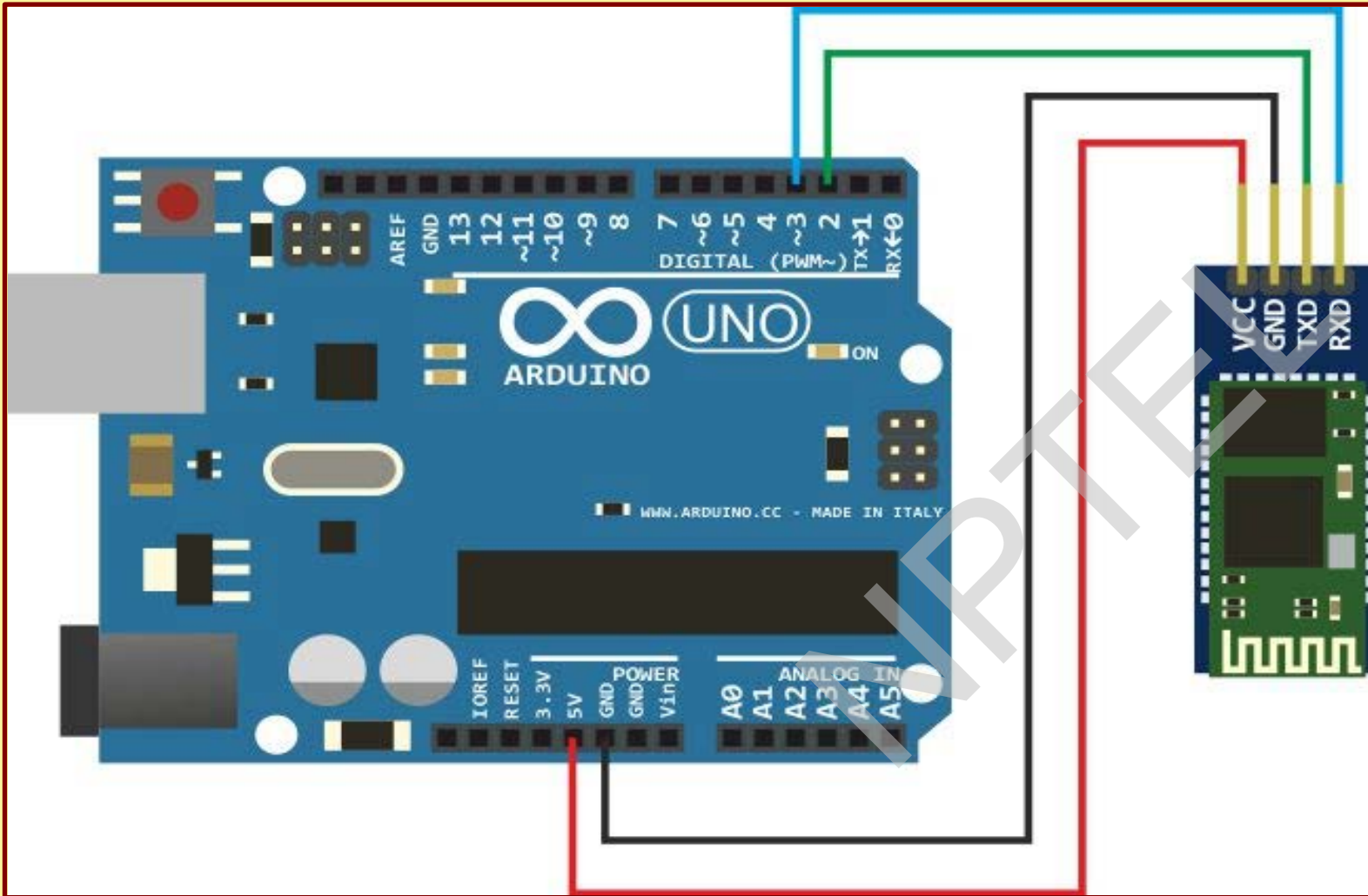
What is Bluetooth?

- A standard for short-range wireless communication for mobile phones, computers, headphones, speakers, and other electronic devices.
- It uses 2.4 GHz frequency for communication, with typical range of 10 meters.
- Typical data transfer rates:
 - Bluetooth 2.0: up to 4 Mbps
 - Bluetooth 3.0: up to 24 Mbps
 - Recent versions also exist ...
- Arduino compatible Bluetooth modules are available for providing communication.
 - Example: HC-05 Bluetooth module

About the HC-05 Bluetooth Module

- It is a class-2 Bluetooth module with serial data interface that can be used as either Master or Slave.
- Technical specifications:
 - 2.4 GHz frequency
 - +3.3V DC power supply
 - Asynchronous speed of 2.1 Mbps (max)





Typical Arduino Interface



NPTEL ONLINE CERTIFICATION COURSES

**Thank
you!**



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

Lecture 36: Design of a Home Automation System

CONCEPTS COVERED

- ☐ About home automation
- ☐ The home automation experiment
- ☐ Demonstration



About Home Automation

- Home automation or smart home is about building automation for a home.
 - Such a system can control lighting, temperature and humidity, entertainment system and appliances.
 - It can also involve home security like access control and alarm systems.
- Typically connects controlled systems through a suitable user interface.
 - Wall mounted terminals, mobile phone application, over the Internet, etc.
- When connected to the Internet, the home devices form an important constituent of the Internet of Things (IoT).

The Experiment

- An automated system in which an electric appliance can be turned on or off remotely by sending a SMS from a mobile phone.
 - A lamp is used for demonstration purpose.
 - The lamp is turned on when the message ON is sent through SMS.
 - The lamp is turned off when the message OFF is sent through SMS.
- Two versions:
 - A normal version where the message can come from any mobile phone.
 - A secured version where the message must come from a particular mobile phone.

Accessories Used in the Experiment

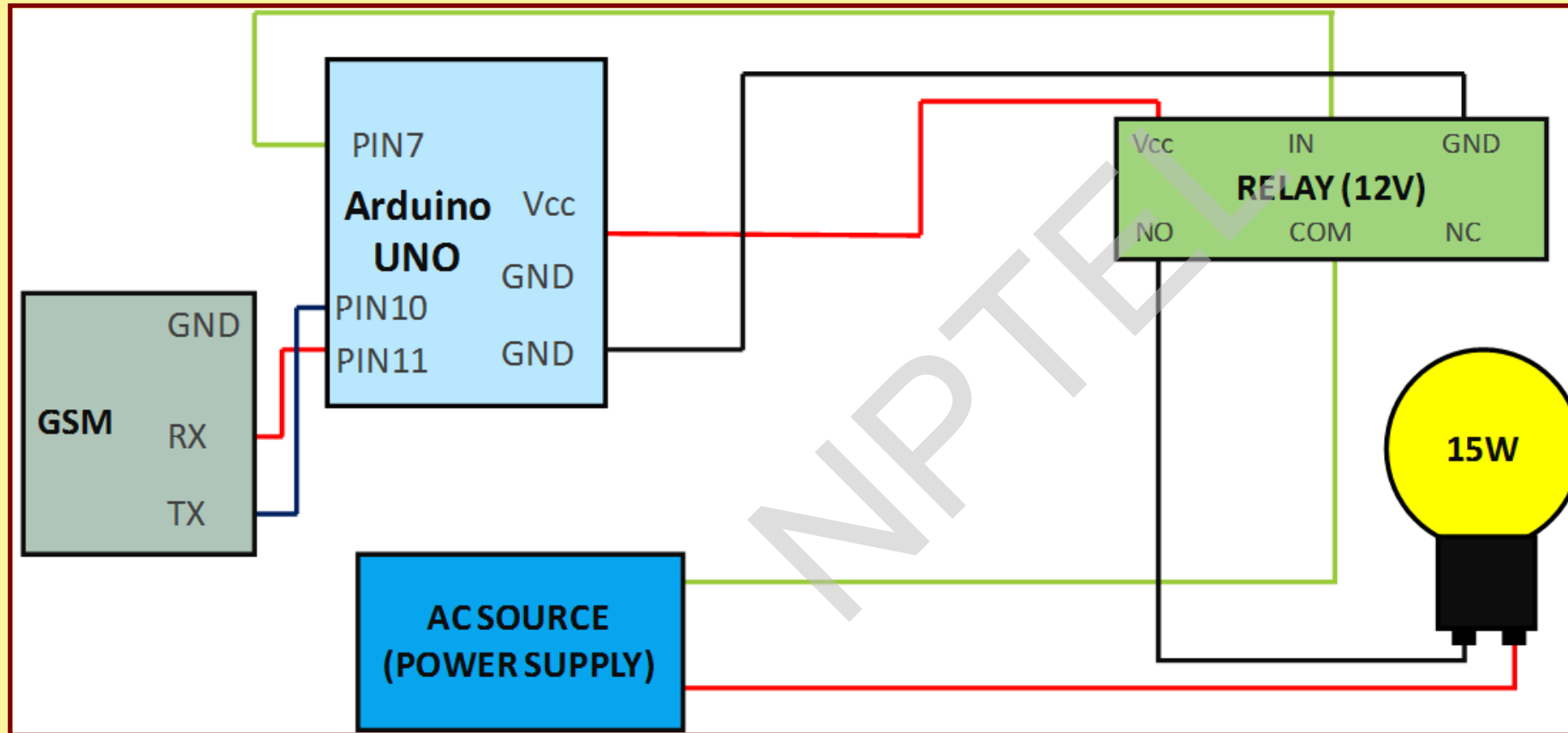
- Microcontroller (Arduino)
 - As the home controller.
- GSM Modem (SIM900A)
 - Used to communicate with the microcontroller from a mobile phone using SMS.
- Relay
 - Actuator device connected to microcontroller to turn ON/OFF an appliance.
- 15W Bulb
 - A home appliance used for demonstration.



Working of the System

- The system polls the GSM modem and waits for arrival of message from the user.
- The microcontroller reads the message, and compares it with the condition for controlling the equipment (viz. the lamp).
- The relay is activated to turn on the lamp when it receives the message ON.
- The relay is activated to turn off the lamp when it receives the message OFF.
- The process continues.
- *For the second version, an additional check is made for the mobile number from where the message has been sent.*

Connection Diagram – Arduino UNO



Arduino C Code (Normal Version)

```
#include <SoftwareSerial.h>
SoftwareSerial SIM900A(10,11);
#define RELAY1 7

int val = 0;
int flag = 0;
int rly = 0;
int len = 0;
```

```
void setup()
{
    pinMode (LED_BUILTIN, OUTPUT);
    Serial.begin (9600);
    pinMode (RELAY1, OUTPUT);
    SIM900A.begin (9600);
    SIM900A.println ("AT+CNMI=2,2,0,0,0");
    delay (100);
}
```

```

void loop()
{
    if (val == 0)
    {
        off();  val = 1;
    }
    while(1)
    {
        readsms();
    }
}

```

```

void on()
{
    digitalWrite(RELAY1,0);  // Turn Relay ON
    Serial.println("Light ON");
    rly = 1;
}

void off()
{
    digitalWrite(RELAY1,1);  // Turn Relay OFF
    Serial.println("Light OFF");
    rly =0;
}

```

```

void readsms() {
    String buffer = readSIM900A();
    if (buffer.startsWith("\r\n+CMT: "))
    {
        Serial.println (buffer);
        Serial.println ("*** RECEIVED AN SMS ***");
        len = buffer.length();
        Serial.println (buffer, substring(len-5, len));
        if (buffer.indexOf("ON")>0 || buffer.indexOf("on")>0
            || buffer.indexOf("oN")>0 || buffer.indexOf("On")>0 )
            on();
        else if (buffer.indexOf("OFF")>0 || buffer.indexOf("Off")>0
            || buffer.indexOf("oFF")>0 || buffer.indexOf("off")>0
            || buffer.indexOf("oFf")>0 || buffer.indexOf("ofF")>0 )
            off();
    }
    buffer.remove (0, len-1);
    delay(100);
    return 0;
}

```

```
String readSIM900A()  
{  
    String buffer;  
    while (SIM900A.available())  
    {  
        char c = SIM900A.read();  
        buffer.concat(c);  
        delay(10);  
    }  
    return buffer;  
}
```

Arduino C Code (Secure Version)

```
#include <SoftwareSerial.h>
SoftwareSerial SIM900A(10,11);
#define RELAY1 7

int val=0;
int flag = 0;
int rly = 0;
int len=0;
```

```
void setup()
{
    pinMode (LED_BUILTIN, OUTPUT);
    Serial.begin (9600);
    pinMode (RELAY1, OUTPUT);
    SIM900A.begin (9600);
    SIM900A.println ("AT+CNMI=2,2,0,0,0");
    delay (100);
}
```

```

void loop()
{
    if (val == 0)
    {
        off();  val = 1;
    }
    while(1)
    {
        readsms();
    }
}

```

```

void on()
{
    digitalWrite(RELAY1,0);  // Turn Relay ON
    Serial.println("Light ON");
    rly = 1;
}

void off()
{
    digitalWrite(RELAY1,1);  // Turn Relay OFF
    Serial.println("Light OFF");
    rly =0;
}

```

```

void readsms() {
    String buffer = readSIM900A();
    if (buffer.startsWith("\r\n+CMT: ")
        && ((buffer.indexOf("+9194xxxxxxxx") > 0)
            || (buffer.indexOf("+9193xxxxxxxx") >
0)))
    {
        Serial.println ("*** RECEIVED AN SMS ***");
        len = buffer.length();
        Serial.println (buffer, substring(len-5, len));
        if (buffer.indexOf("ON")>0)
            on();
        else if (buffer.indexOf("OFF")>0)
            off();
    }
    buffer.remove (0, len-1);
    delay(100);
    return 0;
}

```

```
String readSIM900A()
{
    String buffer;
    while (SIM900A.available())
    {
        char c = SIM900A.read();
        buffer.concat(c);
        delay(10);
    }
    return buffer;
}
```



NPTEL ONLINE CERTIFICATION COURSES

Thank
you!



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

**Lecture 37: Design of a Simple Alarm System
using Touch Sensor**

CONCEPTS COVERED

- ☐ Touch sensor interfacing
- ☐ Experiment using touch sensor and GSM
- ☐ Demonstration

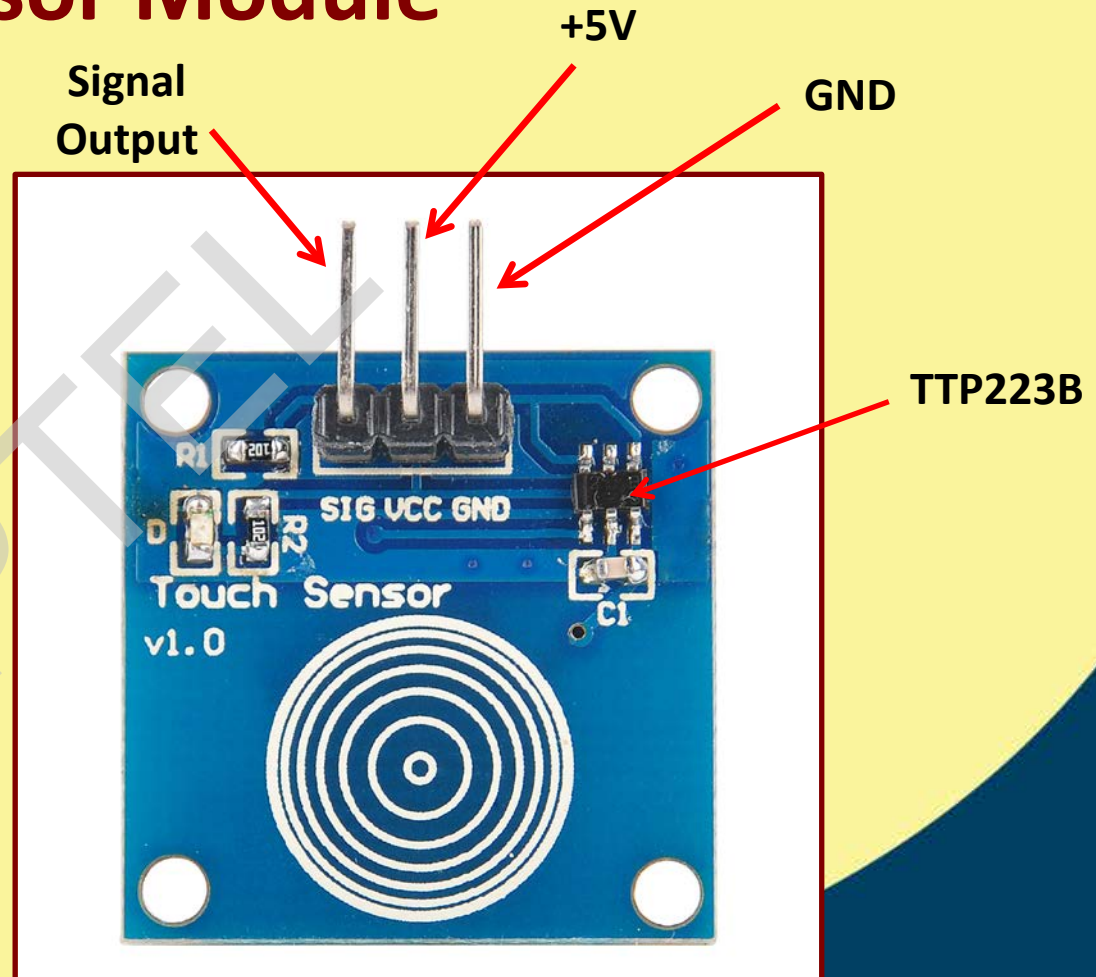


Introduction

- This experiment demonstrates another application of home automation system.
- Related to home security.
 - An automated alarm generation system that warns the owner of the house whenever an unauthorized access takes place.
- How is unauthorized access detected?
 - Through a touch interface circuit, similar to car intrusion alarm systems.
- What happens after the detection?
 - A suitable alarm system can be activated.
 - The owner can be alarmed by sending a SMS on his/her mobile phone.

Interfacing the Touch Sensor Module

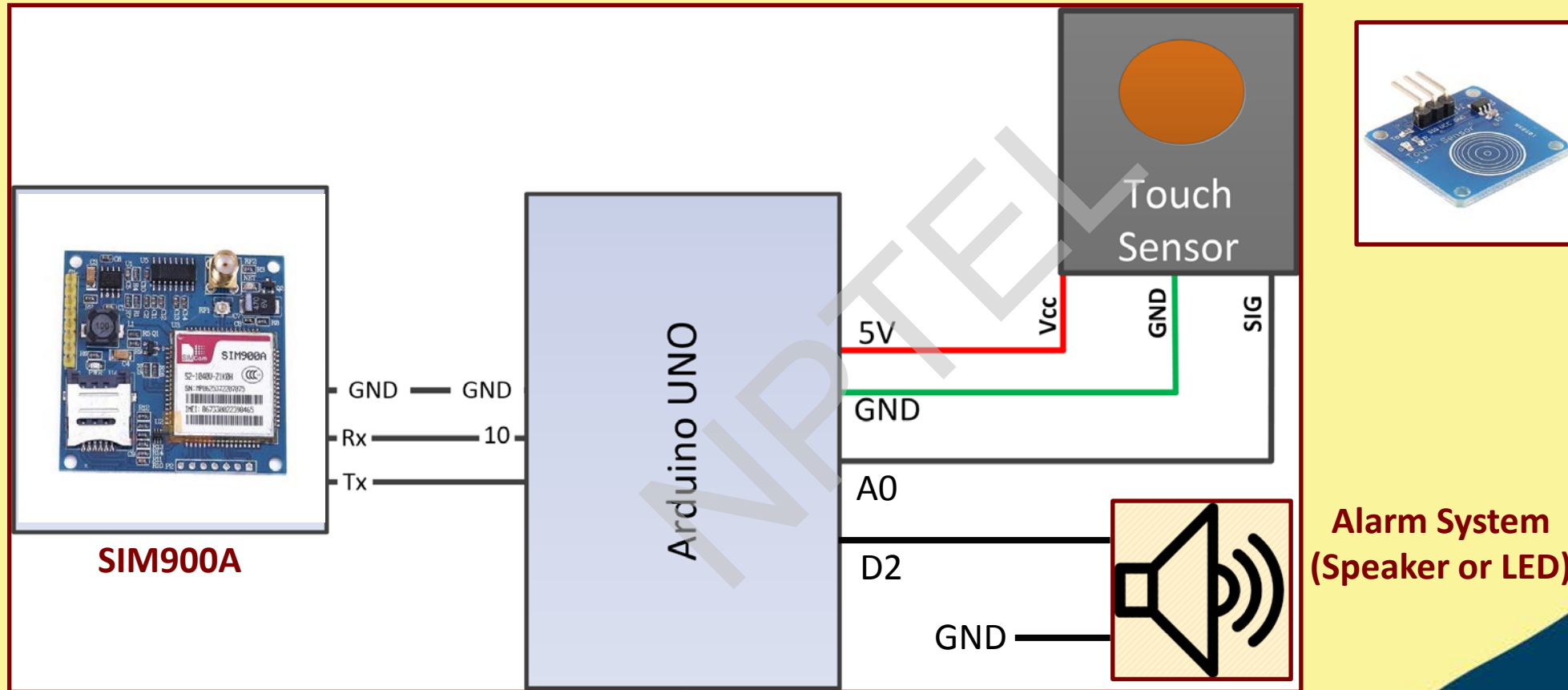
- The touch sensor module contains TTP223B, which is a 1-key touch pad detector IC, and it is suitable to detect capacitive element variations.
- The module has a +5V power supply input, a GND input, and a digital signal output that indicates the presence or absence of touch.



The Experiment

- An Arduino UNO microcontroller system connects to the touch sensor, and a GSM module (SIM900A).
 - The microcontroller continuously polls the touch sensor to find out whether an intrusion (i.e. a touch) has been detected.
 - If so, a suitable message is sent to the person concerned over SMS.
 - An alarm system is also activated.

Connection Diagram



C Code for Arduino UNO

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11);
int analogPin = A0;
int touchValue = 0;

void setup()
{
  mySerial.begin (9600);
  Serial.begin (9600);
  delay (100);
  pinMode (2, OUTPUT);
  digitalWrite (2, HIGH);
}
```

```
void loop()
{
  touchValue = analogRead(analogPin);
  Serial.print (touchValue);
  Serial.print ("\n");
  if (touchValue > 1000)
  {
    SendMessage();
    digitalWrite (2, LOW);
    delay (2000);
    digitalWrite (2, HIGH);
  }
}
```

```
void SendMessage()  
{  
    mySerial.println ("AT+CMGF=1");  
    delay (1000);  
    mySerial.println ("AT+CMGS=\"+919485xxxxxx\"\\r");  
    delay (1000);  
    mySerial.println ("EMERGENCY THEFT ALERT!!!");  
    delay (100);  
    mySerial.println ((char)26);  
    delay (1000);  
}
```



NPTEL ONLINE CERTIFICATION COURSES

**Thank
you!**