

EECS 189

THE CELLULAR REMOTE ACCESS SYSTEM
THE CELLULAR REMOTE ACCESS SYSTEM

EECS 189

Group Members

- Ranjit Sandhu EE (Project Leader & Code Developer)
- Premal Shah EE (Python Script Developer)
- Patrick Krabeepetcharat EE (Hardware Config. & Fabrication)
- Allen Moheimani EE (Research & Presentation)

Faculty

Professor H. Kumar Wickramasinghe

Keram Uenal PhD

EECS 189

Scenario

You Flew to an out-of-state Engineering Conference and Forgot if you locked your car.

What to Do?

Hope nobody notices your new BMW unlocked in LAX

EECS 189

Our Solution

Your Car is Safe.



Remotely Activate Car Alarm System

Design Process

Idea #1

ANALOG CIRCUIT

- Establish Connection then use tone recognition to distinguish which function to perform
- Define Functions (eg. lock and unlock) using logic gates.
- Assign 4 digit code to pre-defined functions

Design Process

Idea #1

Problems Encountered

- Cost of parts
- Security Issues
- Too complex - margin for error high

Design Process

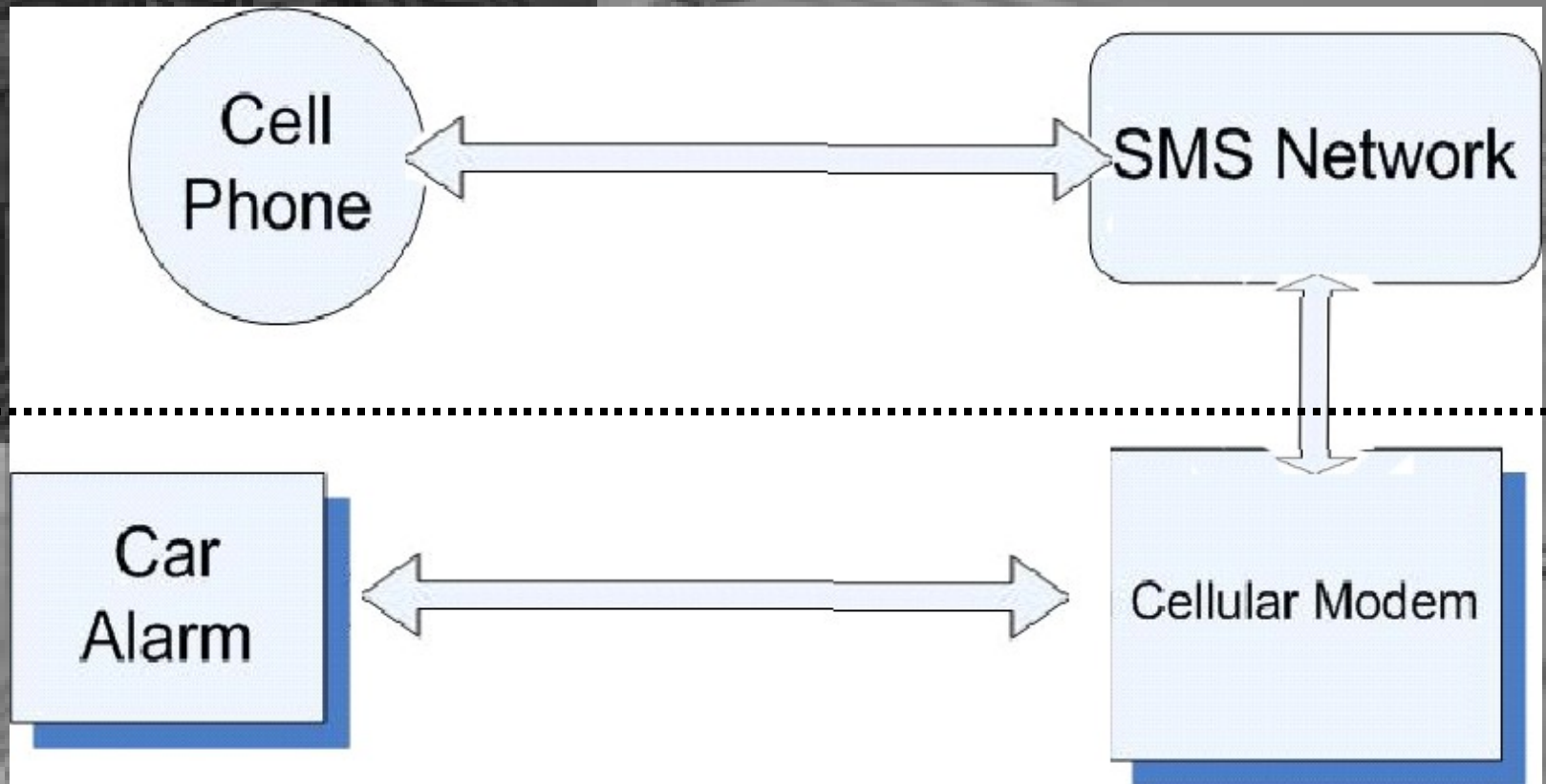
Idea #2

Use Cellular phone to communicate with cellular modem using text messages (SMS)

- GM-862 Quad-PY cellular modem
- Identify which phone numbers to accept commands from
- Scan text message for text command
- Initiate function associated with command

Design Process

Idea #2



Components

- Any SMS Enabled Cellular Phone

• We send pre-defined commands in a text message to the phone number assigned to the SIM card



- Cingular pre-paid SIM Card

Allows connection between user and cellular access system.



Components

Cingular GSM Network Coverage Map

GINGULAR NATION


No Roaming or Long Distance Charges Nationwide

 Cingular Nation (with a GSM handset)

- Anytime, Mobile to Mobile and Night & Weekend Minutes apply

 Future Coverage

- Estimated availability by Year End 2004

 No Service Area



Components

- Quad-band EGSM
850/900/1800/1900 MHz
- TCP/IP stack access via
AT commands
- On Board SIM Holder
- PYTHON Script
Interpreter
- SMS Supported



GM-862 Quad-PY Modem

Components

GM862 Evaluation Board - USB



- We attached the cellular modem to the evaluation board
- The evaluation board was hooked up to LEDs to represent the different states of the car alarm.
- Communication and Power via USB.

Components

- Car alarm system
- Comes as a standard option on almost every car
- Communicates with car
- Activated by Cellular Modem



Alarm System

Python

The programming platform for our project is python. The cellular modem uses Python programs to initiate AT commands that activate specific pins on the cellular module. These commands are received via SMS (text messages).

Python Script

Code

```
import MDM
import MOD
import GPIO

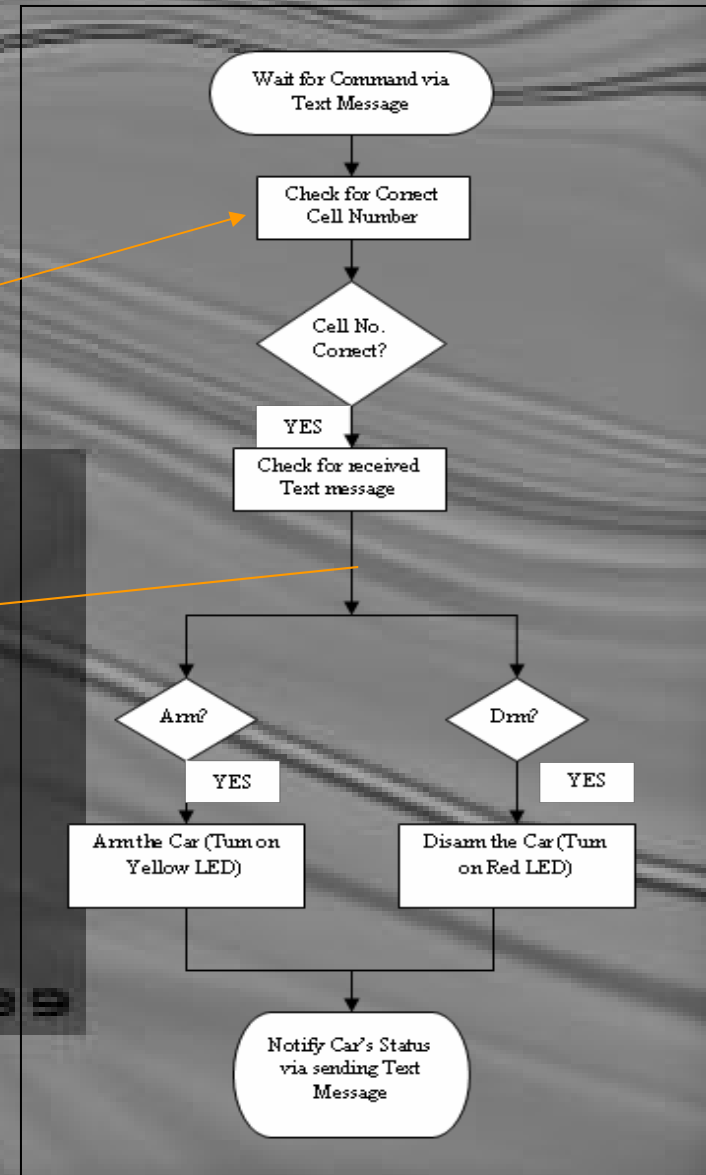
MOD.sleep(150)

a = MDM.send('AT#BND=1\r', 90)    #set to north american frequencies
a = MDM.receive(50)

a = MDM.send('AT+CMGF=1\r', 90)   #set sms to text mode
a = MDM.receive(50)

i=1
while i<10:
    f = MDM.send('AT+CPMS=SM\r', 90)
    g = MDM.receive(50)
    if (g.find('5,30,5,30,5,30') != -1):
        b = MDM.send('AT+CMGR=5\r', 90)
        c = MDM.receive(50)
        if (c.find('19099577528') != -1):
            if (c.find("on") != -1):
                GPIO.setIOWrite(3,0)
                GPIO.setIOWrite(7,1)
                d = MDM.send('AT+CMSS=1\r', 90)
                d = MDM.receive(60)
                e = MDM.send('AT+CMGD=5\r', 90)
                e = MDM.receive(60)
            elif (c.find("off") != -1):
                GPIO.setIOWrite(7,0)
                GPIO.setIOWrite(3,1)
                d = MDM.send('AT+CMSS=2\r', 90)
                d = MDM.receive(60)
                e = MDM.send('AT+CMGD=5\r', 90)
                e = MDM.receive(60)
            else:
                d = MDM.send('AT+CMSS=3\r', 90)
                d = MDM.receive(60)
                e = MDM.send('AT+CMGD=5\r', 90)
                e = MDM.receive(60)
        else:
            d = MDM.send('AT+CMSS=4\r', 90)
            d = MDM.receive(60)
            e = MDM.send('AT+CMGD=5\r', 90)
            e = MDM.receive(60)
    else:
        #GPIO.setIOWrite(3,1)
        #GPIO.setIOWrite(7,1)
        MOD.sleep(100)
```

Flow Chart



AT Commands in Python

Code

Hyper Terminal

```
import MDM
import MOD
import GPIO

MOD.sleep(150)

a = MDM.send('AT#BND=1\r', 90)    #set to north american frequencies
a = MDM.receive(50)

a = MDM.send('AT+CMGF=1\r', 90)   #set sms to text mode
a = MDM.receive(50)

i=1
while i<10:
    f = MDM.send('AT+CPMS=SM\r', 90)
    g = MDM.receive(50)
    if (g.find('5,30,5,30,5,30') != -1):
        b = MDM.send('AT+CMGR=5\r', 90)
        c = MDM.receive(50)
        if (c.find('19099577528') != -1):
            if (c.find("on") != -1):
                GPIO.setIOvalue(3,0)
                GPIO.setIOvalue(7,1)
                d = MDM.send('AT+CMSS=1\r', 90)
                d = MDM.receive(60)
                e = MDM.send('AT+CMGD=5\r', 90)
                e = MDM.receive(60)
            elif (c.find("off") != -1):
                GPIO.setIOvalue(7,0)
                GPIO.setIOvalue(3,1)
                d = MDM.send('AT+CMSS=2\r', 90)
                d = MDM.receive(60)
                e = MDM.send('AT+CMGD=5\r', 90)
                e = MDM.receive(60)
            else:
                d = MDM.send('AT+CMSS=3\r', 90)
                d = MDM.receive(60)
                e = MDM.send('AT+CMGD=5\r', 90)
                e = MDM.receive(60)
        else:
            d = MDM.send('AT+CMSS=4\r', 90)
            d = MDM.receive(60)
            e = MDM.send('AT+CMGD=5\r', 90)
            e = MDM.receive(60)

    else:
        #GPIO.setIOvalue(3,1)
        #GPIO.setIOvalue(7,1)
        MOD.sleep(100)
```

```
PREM - HyperTerminal
File Edit View Call Transfer Help
[Icons]
Disconnected Auto detect Auto detect SCROLL CAPS NUM Capture

AT
OK
AT#BND=1                % NORTH AMERICAN FREQUENCY

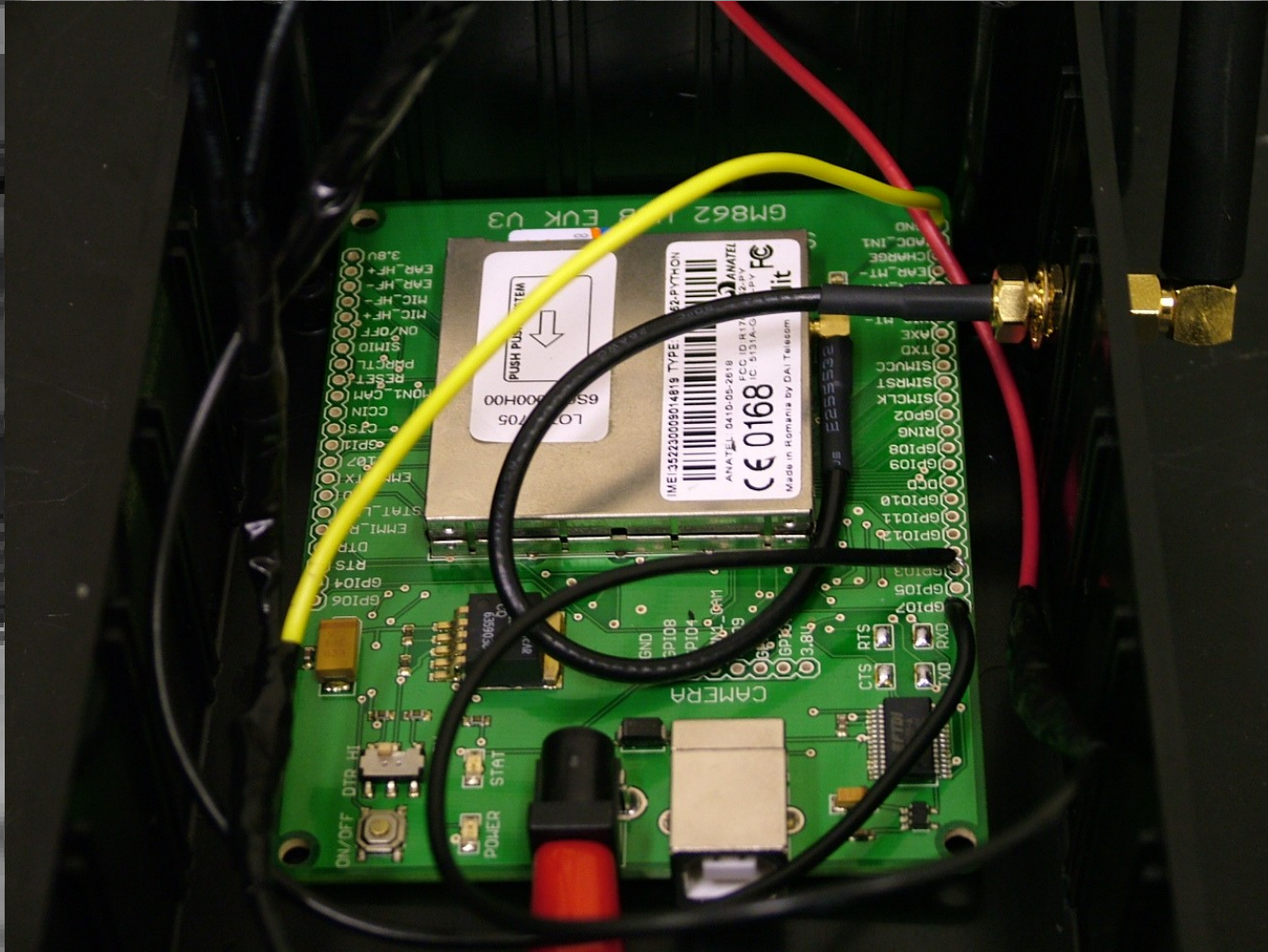
AT#GPIO = 7,0,1        % TURN ON GPIO 7
OK
AT#GPIO = 7,0,0        % TURN OFF GPIO 7
OK
AT+CMGD = 1            % DELETING SMS FROM LOCATION '1'
OK
AT+CMGR = 2            % READING A SMS FROM LOCATION 2
OK
```




Video Simulations

EECS 189

Images



Images



Problems Encountered

- Lack of programming experience
 - Back order on parts
 - Lack of resources
 - Timing issues
 - Debugging issues

Possible Expansions

- Additional Functions: turning on the car, opening the trunk
- Additional Functions
 - C-MOS Camera

Cost

Manufacturing

Parts: \$ 320.00

Salary: \$18000.00

Engineering Hours: 150

Hourly Rate: \$30.00

of Employee: 4

Total: \$18320.00

Retail Price: \$ 200.00

+ (Cellular Service
charge \$10 - \$30 per month.)

Personal

GM 862 Modem: \$ 90.00

EVK Board: \$130.00

Microcontroller: \$ 50.00

Sim Card (Cing.): \$ 25.00

Misc. (LEDs & : \$ 25.00

Antenna, Packaging):

Total: \$ 320.00



QUESTIONS?

EECS 189