

## IOT with SMS

**Author: U. Schiara**

**Date: 06/01/2020**

**Rev.1**

## TABLE OF CONTENTS

1. Foreword .....	3
2. Scope .....	3
3. The Gateway.....	3
4. The Gateway Software .....	3
a. The overall structure of the Gateway app.....	4
b. Control Flow .....	5
5. The Arduino board.....	6
6. The Arduino software .....	6
a. Control Flow .....	6
7. Next Steps.....	6
Figure 1 .....	3
Figure 2 .....	4
Figure 3 .....	4
Figure 4 .....	6

## 1. Foreword

IOT is increasing every day and the use of a remote communication to manage the house's devices is a "nice to have" for everybody who speaks a little about bits and bytes, and electronics. The use of Internet is therefore becoming higher by people who want to be up to date, but this also increase the possibility to make our home routers to be hacked. Nevertheless there is another possibility to connect remotely our homes : by using the "old" SMS's. Sometimes the "data mode" of our smartphones is not working, due to lack of signal, while the SMS is still working, therefore allowing us to communicate without the need of the Internet. In addition to this, an SMS is composed by 128 bytes, and if you consider that each byte is composed by 8 bits, supposing to map an ON/OFF command on any bit of 100 bytes (leaving for example 28 bytes for a password) you can drive up to 800 actuators. Or, if you want to drive analogue actuators like a fan speed, or a temperature controller, you can allocate single bytes (0-255) for each actuator: at the end it depends on your needs and your "fantasy". The protocol for command transmission is up to you; just remember: the simplest, the best. Last, but not least, by using SMS's you don't need to allocate a fixed IP to your home router, which is granted only if you apply to a dedicated service provider.

## 2. Scope

The aim if this document is to describe how to use a combination of a MIT AI2 app on your smartphone and an Arduino board to do the job: to send an SMS command by your smartphone, to receive it by another smartphone that acts as a BT gateway to an Arduino board the actuates the commands. The following drawing explains how:

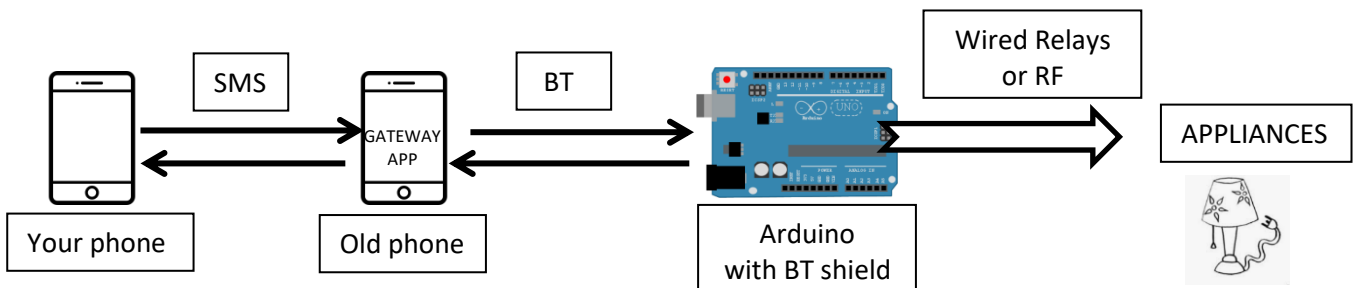


Figure 1

## 3. The Gateway

Smartphones manufacturers put on the market new models every month, or even more often, so our kids put and forget the "old" ones, though still working, down in a corner of the drawer of their desk. To make them becoming useful again, gives them the opportunity of a second life. So take them out of the drawer and follow the rest of this document.

## 4. The Gateway Software

By loading on the Gateway a MIT AI2 developed app, this smartphone receives the SMS by a sender, and it forwards it, transparently, to the Arduino board equipped with a BT shield. The Arduino board then analyzes the received SMS and sends back to the Gateway the appropriate response. The gateway on its turn, sends the response to the sender by means of an SMS. The loop is then closed: the user receives the information about the accomplishment (or failure) of the command sent. The Gateway does not perform any check on what it receives.



**THE GATEWAY USER INTERFACE**

The gateway user interface provides:

- The gateway status
- A connect button to connect the BT (when the automatic connection fails)
- The exit button, to shut down the app
- A field where to insert the phone number to dial to for the answer
- The Phone Number save button (it writes on the phone memory for later retrieval)
- A field where the message sent by Arduino (three rows available) is displayed
- A field where the SMS sent by the caller is displayed

Figure 2

**a. The overall structure of the Gateway app.**



Figure 3

Where:

Clock2: to connect the BT shield, it fires 2 seconds after being enabled by Pro\_Conn

Clock3: similar to a watchdog, it fires every 10 seconds, if not retriggered by Clock4.

Clock4: fires every 200 milliseconds in order to catch the data incoming from the BT. It retriggers Clock3 if it receives a "live" from Arduino. Arduino must therefore send one character before 10 seconds after the last one. Typically every two or three seconds Arduino must send one character to retrigger Clock3. If the message received by the BT is complete, and the correct trigger sequence begins the message, it is sent via SMS to the caller phone.

Pro\_Conn: request of connection, it enables Clock2

Try\_Conn: performs a connection to the specified BT address written into the block itself. If necessary, this portion of code can be improved by reading the BT address from a file, for example.

Texting1\_MessageReceived: whenever a SMS is received, it forwards it to the BT line.

### **b. Control Flow**

The program flow is:

1. Screen initialize:
  - initialize the Clocks periods but do not enable (Clock2 = 2 s; Clock3=10 s; Clock4 = 200 ms)
  - read the telephone number from a file
  - keeps screen on
  - write "Connecting
  - start Clock2 for automatic BT connection
  - send a trigger character to Arduino to start its operations
  - wait for events
2. Wait for events:
  - Clock2 active only once, fires 2 s after activation, waits for BT connection
  - Clock3 every 10 s if not retriggered by clock2
  - Clock4 every 200 ms

## 5. The Arduino board

This board receives the SMS command by the Gateway by means of the BT communication, it analyzes the SMS contents and, if it is a valid command (i.e. password+command), it execute the actuation required, and sends back on the BT line the answer. The Gateway receives the answer and forwards it in SMS format to the sender phone. The Arduino board then can drive a relays box or, if equipped with a RF shield (like the NRF24L01) you can remotely drive actuators spread in your house (in this case the RF receiver shall be used in conjunction with an Arduino Nano, for example, or a bare bone processor, to be capable to understand and actuate the received command).

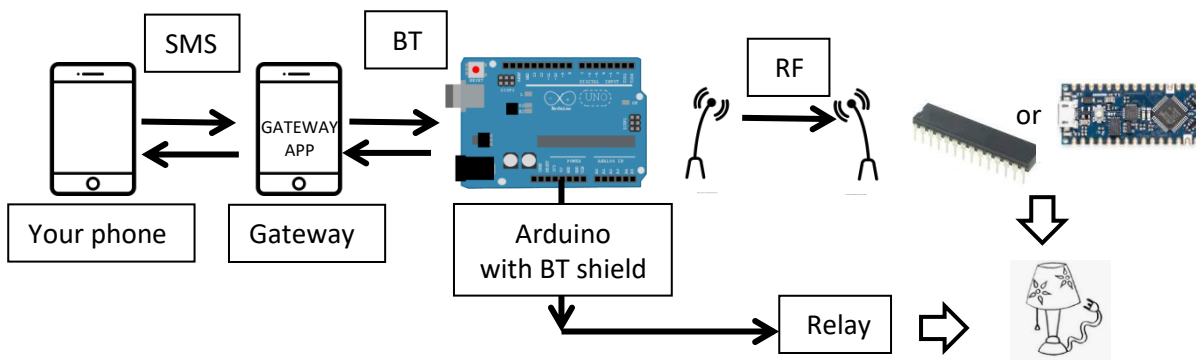


Figure 4

## 6. The Arduino software

The software running on Arduino shall be capable of receiving the SMS sent by the gateway on the BT communication channel, to analyze it, to actuate (or to retransmit on RF the actuation command) and to transmit the feedback on the BT line.

### a. Control Flow

The program flow is:

- Initialization
- wait for the trigger character
- send a "ready" message
- enter the neverending loop
- in case of BT message received, actuate its command, wait a while to allow the Gateway to be ready to accept the answer (typ. 1 s)
- reply on BT with a feedback
- every two seconds, send a "live" character to the app, so to retrigger the watchdog (Clock3)

## 7. Next Steps

To implement the RF network.